



## A New Efficient Hybrid Approach for Machine Learning-Based Firefly Optimization

Mohamed Alsoul<sup>1</sup>, Hegazy Zaher<sup>2</sup>, Naglaa Ragaa<sup>1</sup>, Eyman Mostafa Oun<sup>1</sup>

<sup>1</sup> Department of Operations Research and Management, Faculty of Graduate Studies for Statistical Research, Cairo University, Cairo, Egypt

<sup>2</sup> Department of Mathematical Statistics, Faculty of Graduate Studies for Statistical Research, Cairo University, Cairo, Egypt.

Received: 12/8/2022

Accepted: 7/10/2022

Published: 30/9/2023

### Abstract

Optimization is the task of minimizing or maximizing an objective function  $f(x)$  parameterized by  $x$ . A series of effective numerical optimization methods have become popular for improving the performance and efficiency of other methods characterized by high-quality solutions and high convergence speed. In recent years, there are a lot of interest in hybrid metaheuristics, where more than one method is ideally combined into one new method that has the ability to solve many problems rapidly and efficiently. The basic concept of the proposed method is based on the addition of the acceleration part of the Gravity Search Algorithm (GSA) model in the Firefly Algorithm (FA) model and creating new individuals. Some standard objective functions are used to compare the hybrid (FAGSA) method with FA and the traditional GSA to find the optimal solution. Simulation results obtained by MATLAB R2015a indicate that the hybrid algorithm has the ability to cross the local optimum limits with a faster convergence than the luminous Fireflies algorithm and the ordinary gravity search algorithm. Therefore, this paper proposes a new numerical optimization method based on integrating the properties of the two methods (luminous fireflies and gravity search). In most cases, the proposed method usually gives better results than the original methods individually.

**Keywords:** Firefly Algorithm (FA), Gravitational Search Algorithm (GSA), Machine Learning (ML), Numerical Optimization Method.

### تقنية مركبة لتعلم الآلة التي تركز على التقنية المثالية للبراعات المضيفة

محمد الصول<sup>1</sup>، حجازي زاهر<sup>2</sup>، نجلاء رجاء<sup>1</sup>، إيمان مصطفى<sup>1</sup>

<sup>1</sup> قسم بحوث وإدارة العمليات، كلية الدراسات العليا للبحوث الإحصائية، جامعة القاهرة، القاهرة، مصر.

<sup>2</sup> قسم الإحصاء، كلية الدراسات العليا للبحوث الإحصائية، جامعة القاهرة، القاهرة، مصر.

### الخلاصة:

إن الأمثلية هي مهمة تصغير (تدنى) أو تعظيم دالة الهدف (دس) ذات المعلمة س. وقد ذاع صيت سلسلة من طرق الأمثلية الرقمية الفعالة بسبب تحسينها أداء وكفاءة الطرق الأخرى التي تتميز بإيجاد حلول عالية الجودة وتقارب عالي السرعة. وفي السنوات الأخيرة، كان هناك الكثير من الاهتمام بعلم الاستدلال الفوقى المركب

(الهجين) ، حيث يتم دمج أكثر من طريقة، بشكل مثالي، في طريقة واحدة جديدة تمتلك القدرة على حل العديد من المشكلات بسرعة وكفاءة. إن المفهوم الأساسي للطريقة المقترحة مبني على إضافة جزء التسارع من نموذج خوارزمية بحث الجاذبية (GSA) الى نموذج خوارزمية اليراعات المضيئة (FA). تُستخدم بعض دوال الهدف القياسية في مقارنة الطريقة المركبة مع كل من خوارزمية اليراعات المضيئة FA و خوارزمية بحث الجاذبية GSA التقليدية للعثور على الحل الأمثل. تشير نتائج المحاكاة التي تم الحصول عليها بواسطة برنامج ماتلاب ر 2015 إلى أن الخوارزمية المركبة ذات قدرة على تخطي الحدود المحلية المثلى بتقارب أسرع من خوارزمية اليراعات المضيئة وخوارزمية بحث الجاذبية العادية. في معظم الحالات ، تعطي الطريقة المقترحة نتائج أفضل من تنفيذ الطرق الأصلية بشكل فردي.

## 1. Introduction

In machine learning (ML), classical optimization algorithms fail to give an appropriate solution to optimization problems with a high-dimensional search space, which grows exponentially with the problem size. Therefore, utilizing accurate techniques (such as exhaustive search) is recommended when exhaustive testing of these issues is impractical[1]. Hybrid approaches can be used to speed up the process of finding a satisfactory solution and improve the sustainability of energy production infrastructures in many areas such as households. Several metaheuristic hybridization methods are presented in ([2] and [3]). As a result, two or more algorithms can be hybridized as homogeneous or heterogeneous in high-level or low-level systems via a relay or convolutionary methods.

Recently, several heuristic evolutionary algorithms have been enhanced. These algorithms include many different ones, such as the Bat Algorithm (BT)[4], Cuckoo Search (CS)[5], Genetic Algorithm (GA)[6], Firefly Algorithm (FA)[7] and [8], Differential Evolution (DE)[9], Gravitation Search Algorithm (GSA)[10], Ant Colony (AC)[11], Particle Swarm Optimization (PSO)[12] and [13], and Artificial Bee Colony (ABC)[14]. The main common objective for all of them is to search for and find a better solution from all available inputs. A heuristic algorithm should have two or more essential characteristics to ensure that a global optimum is found.

According to [15], exploration and exploitation are the two fundamental aspects that play an important role in describing the working of any algorithm. Exploration is an algorithm's ability to search for entire parts of a problem in dimension space, whereas exploitation is the algorithm's ability to find the near optimum solution. All heuristic optimization algorithms strive to find a global optimum by efficiently keeping a balance between exploitation and exploration abilities. The method allows controlling the balance between investigating unknown regions of the dimension space (exploration) and investigating further known regions of dimension space to have low-energy structures (exploitation) [16].

Because of the aforementioned limitations, current heuristic optimization algorithms can only solve a limited set of problems. That is, there is no algorithm capable and sufficient to find a solution for all optimization problems [17]. Therefore, a hybrid optimization algorithm is a way to make a balance between comprehensive exploration and exploitation when creating a new creative algorithm. Due to its simplicity, speed of convergence, and ability to search for a global optimum, in hybrid approaches, FA is one of the most extensively utilized evolutionary algorithms.

There have been various studies done in the literature to aggregate FA with other algorithms such as hybrid FireFly Particle Swarm Optimization (FFPSO)[18], where the main feature of

the FA algorithm is the attractiveness of brightness intensity that models the optimal solution and better exploitation due to the absence of the velocity vector (V) and personal best (pbest) terms in it, while PSO ([12] and [13]) is trying to take advantage of two characteristics to reach the best velocity; the first is based on the best personal position (pbest) and the second is the best global flight position (gbest) [19], Firefly Algorithm Differential Evolution (FADE) enhances searching efficiency by executing FA and DE in parallel to promote information sharing among the Firefly population [20], and Hybrid FireFly Algorithm and Cuckoo Search (HFFACS) balances exploration and exploitation by its simplicity and low computational cost when applied to a wide range of problems [21]. These hybrid algorithms are designed to reduce the likelihood of being trapped in a local optimum. GSA, a novel heuristic optimization method was recently introduced [22]. This study introduces a new hybrid model that combines FA and GSA algorithms, named FAGSA, and uses twenty-three benchmark functions to compare the performance of the heuristic algorithm of the FA and the GSA algorithm with the new hybrid algorithm, FAGSA.

Instead of using traditional feature extraction and optimization techniques, several metaheuristic techniques are developed to solve the problem of feature optimization, such as the Firefly Sequential quadratic programming (FaSqp) algorithm introduced by [23], the FFPSO algorithm introduced by [18], and the FADE algorithm introduced by [20]. For real-world optimization problems, meta-heuristic algorithms have been proven to outperform gradient-based techniques. According to [24], the firefly algorithm has one of the optimization algorithms that can deal with multimodal functions more naturally and efficiently. Fireflies are created artificially and deployed randomly in decision space. Then, each firefly emits a flashing mechanism for other fireflies. This paper introduces the metaheuristic FAGSA as a feature extraction method. The FA and GSA sections contain a detailed formulation and an explanation of each algorithm.

### 1.1. Standard Firefly Algorithm

FA is an evolutionary computation algorithm proposed by X.S. Yang in 2007. The firefly algorithm was developed by simulating firefly brightness (mating) behavior. Although this algorithm is similar to Particle Swarm Optimization (PSO) ([12], and [13]), Artificial Bee Colony (ABC) Optimization[14], and Ant Colony Optimization (ACO)[11] were significantly easier to implement [21].

Fireflies are small insects that emit a bright light intended to attract other fireflies and also hunt prey [25]. They emit a short series of light flashes in a repetitive pattern. From elementary physics, it is clear the intensity of light  $I$  is inversely proportional to the square of the distance  $r$  from the source, where the attractiveness  $I$  of fireflies decreases as the distance  $r$  increases. Hence, most fireflies can only be seen from a few hundred meters away. The implementation of the fitness function depends on the behavior of the brightness of the light emitted by the fireflies to implement this algorithm. For the aim of plainness, it is assumed that firefly light intensity attractiveness is specified by its brightness  $I$ , which correlates with the fitness function.

#### a. Attractiveness and light intensity

The brightness of a firefly is determined from the encoded objective function depending on the equations below.

$$\text{Min } f(\mathbf{x}_i) \quad \mathbf{X} = (x_1, \dots, x_d)^T \quad (1)$$

where

$X$  is the population size at a given position  $r$ ,

$I$  is the intensity of the light reflected by the source, which can be calculated using the firefly's total brightness as in  $I(r)$ , which is proportional to fitness. As a result,

$I(r)$  follows the inverse square law, which is defined as:

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

The variation in firefly attractively  $\beta$  with nearby others by distance  $r$  is proportional to  $I(r)$  and can be defined as follows:

$$\beta(r) = \beta_0 e^{-\gamma r^2}. \tag{3}$$

Where  $\beta_0$  is an attractiveness at  $r = 0$  based on brightness,  $\gamma$  is the fixed light absorption coefficient and theory,  $\gamma \in [0, \infty)$ , but in a traineeship  $\gamma=O(1)$  is determined by the characteristic length  $r$  of the system to be optimized and in most applications, it typically varies in range from 0.01 to 100.

$r$  is the distance between fireflies at position  $x_i, x_j$  and estimate the using following formula

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^d (x_{i,k}(t) - x_{j,k}(t))^2}, \tag{4}$$

where  $d$  is a dimension in space,  $x_i$  and  $x_j$  are two fireflies and  $j > i$ .

The movement of a firefly  $i$  is attracted to another more attractive (brighter) firefly  $j$  is determined by

$$x_i(t + 1) = x_i(t) + \beta_0 e^{-\gamma r^2} (x_j(t) - x_i(t)) + \alpha \epsilon_i \tag{5}$$

where  $x_i(t)$  is the current position of a firefly.

$\beta_0 e^{-\gamma r^2} (x_j(t) - x_i(t))$  is the attractiveness ( $\beta$ ) of a firefly (the attraction towards neighbors  $x_j$ ),  $\alpha \epsilon_i$  indicates the random walk of a firefly with  $\alpha \in [0, 1]$  being the randomization parameter and  $\epsilon_i$  is Gaussian or uniformly distributed in  $[0, 1]$ . If the scales differ significantly in different dimensions such as  $-105$  to  $105$  in one dimension while, say,  $-0.001$  to  $0.01$  along with the other, it is recommended to replace  $\alpha$  with  $\alpha S_k$  where the scaling parameters are  $S_k$  ( $k = 1, \dots, d$ ).

If there is no brighter firefly  $j$   $\beta_0 = 0$ , it will move randomly (simple random walk).

$$x_i(t + 1) = x_i(t) + \alpha (rand - \frac{1}{2}). \tag{6}$$

The pseudo-code for the firefly algorithm

Algorithm 1- FA [7]

```

1:   Define the objective function  $f(x)$ ,  $X = (x_1, x_2, \dots, x_d)^T$ 
2:   Set default values for randomization coefficient  $\alpha$ , firefly attractiveness  $\beta_0$ , media light absorption coefficient  $\gamma$ , population size  $P$ , and maximum generation number MGN.
3:   Generate random numbers for populations  $x_i, i = \{1, 2, \dots, P\}$  {Initialization}
4:   Evaluate the fitness function  $f(x)$  of all solutions in the population
5:   Based on the light intensity of fireflies  $I_i$  at  $x_i$ , calculate the fitness of the initial population  $f(x_i)$ .
6:   for  $t$  in range(1, MGN)
7:       for  $i$  in range( $I, P$ ) (fireflies)
8:           for  $j$  in range( $I, P$ )
9:               by using Eq. 2, compute the light intensity ( $I$ ).
10:              by using Eq. 4, computing the distance between two fireflies
11:              if  $I_i^{(t+1)} < I_j^{(t+1)}$ 
12:                  by using Eq. 5, move firefly  $I_i$  towards firefly  $I_j$ 
13:              else
14:                  else by using Eq. 6, firefly  $i$  is moved randomly towards firefly  $j$ 
15:              end
16:              Obtain attractiveness  $\beta$ , where  $\beta(r) = \beta_0 e^{-\gamma r^2}$ 
17:              Evaluate the fitness function  $f(x_i)$  of all population solutions
18:              Update light intensity  $I$ 
19:          end
20:      end
21:      Sort the solution results and keep the best solutions that exist from the population so far.
22:  end

```

1.2. Standard Gravitational Search Algorithm

The authors in [10] and [26] have proposed a novel heuristic optimization method, GSA. According to Isaac Newton's law of gravitation, every particle in the universe attracts another with a gravitational force, which is directly proportional to the product of their masses and inversely proportional to the square of the distance between them. The following is a mathematical model of the GSA. Assume there are N particles in a system. The process begins by distributing all particles in the search space at random. The gravitational forces exerted by particle j on particle i at any given time are defined as follows [10]:

$$F_{ij}^d(t) = G(t) \frac{M_{pi}(t) * M_{aj}(t)}{R_{ij}(t) + \epsilon} (x_j^d(t) - x_i^d(t)). \tag{7}$$

The active gravitational mass associated with particle j is denoted by  $M_{aj}$ , the passive gravitational mass associated with particle i is denoted by  $M_{pi}$ , the gravitational constant at time t is denoted by G(t),  $\epsilon$  is a small constant, and the Euclidian distance between two particles i and j is denoted by  $R_{ij}$ :

$$R_{ij}(t) = \|x_i(t), x_j(t)\|_2 \tag{8}$$

G(t) is calculated as follows:

$$G(t) = G_0 * \exp(-\alpha * \text{iter}/\text{maxiter}) \tag{9}$$

Where  $\alpha$  and  $G_0$  represent the descending coefficient and initial value,  $iter$  represents the current iteration, and  $maxiter$  represents the maximum number of iterations. The total force acting on particle  $i$  in a problem space with dimension  $d$  is calculated as follows:

$$F_i^d(t) = \sum_{j \in kbest}^N \alpha_j F_{ij}^d(t) \text{ where } i \neq j \text{ and } \alpha_j = [0,1] \tag{10}$$

$$kbest = \left( final_{per} + \left( 1 - \frac{t}{T} \right) \times (100 - final_{per}) \right) \times \frac{N}{100}, final_{per} = 2 \tag{11}$$

In Newton's law of motion, the acceleration of an object is caused by a net force inversely proportional to its mass, so all the acceleration of particles is computed by the following equation:

$$aC_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)} \tag{12}$$

In the preceding equation (12),  $t$  represents a specific time and  $M_i$  represents the mass of the object  $i$ . The following equations are used to calculate the velocity and position of the objects:

$$vel_i^d(t + 1) = \varepsilon_i vel_i^d(t) + \alpha C_i^d(t) \tag{13}$$

where  $\varepsilon$  is uniformly distributed in the interval  $[0, 1]$ .

$$x_i^d(t + 1) = x_i^d(t) + vel_i^d(t + 1) \tag{14}$$

In each round, Equations (15) and (16) are used to update the velocity as well as the mass of each agent  $i$ .

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)}, \tag{15}$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^k m_j(t)}, \tag{16}$$

Firstly, random values are assigned to GSA masses. Each mass is a potential solution. Following the initialization step, the velocities of all masses are defined using equations (13). Meanwhile, the gravitational constant total forces and accelerations are computed as (Eq. 8, 9, 10), respectively. The mass position value is computed using (14). Finally, the GSA will be terminated once the meeting criteria are met.

### The pseudo-code for the Gravitational Search Algorithm

---

**Algorithm 2-** GSA [10]

---

- 1: Establishing the local search space
- 2: parameter initialization,  $X_i = (x_i^1, x_i^2, x_i^3, \dots, x_i^n)$ , for  $i=1, 2, 3, \dots, M$
- 3: Object (agent) fitness evaluation
- 4:  $Kbest$  (Eq. 11),  $G(t)$  (Eq. 9), and  $M_i(t)$  (Eq. 16) should be updated, for  $i = 1, 2, \dots, M$ .
- 5: Compute the total force in different directions (Eq. 10).
- 6: Determine the acceleration and velocity (Eq. 12, 13).
- 7: Updating objects' positions while producing  $x_i^d(t + 1)$ ,  $i = 1, 2, \dots, M$  (Eq. 14).
- 8: Steps 3–7 must be repeated until the stopping criterion is met.

### 1.3. Proposed approach: FAGSA Algorithm

In this section, the acceleration part of the GSA algorithm is used in the FA algorithm, which is similar to previous work in ([18], [20] and [23]). Such integration has the advantage of being better in convergence. Besides, it prevents falling into the local minimum. The FA algorithm has an advantage over other optimization techniques in its simplicity; only a few parameters

need to be modified. The FAGSA algorithm works in the same way as the FA algorithm, except that the position vector of the FA algorithm is modified as the following equation:

$$x_i(t+1) = x_i(t) + \beta_0 e^{-\gamma r^2} (x_j(t) - x_i(t)) + aC_i^d(t) + \alpha \epsilon_i \quad (17)$$

If there is no brighter,  $\beta_0 = 0$  and the firefly moves randomly (simple random walk):

$$x_i(t+1) = x_i(t) + aC_i^d(t) + \alpha \epsilon_i, \quad (18)$$

The pseudo-code for the proposed algorithm

---

**Algorithm 3- FAGSA**

---

- 1: Define the objective function  $f(x)$ ,  $X = (x_1, x_2, \dots, x_d)^T$
- 2: Set the initial values of randomization parameter  $\alpha$ , firefly attractiveness  $\beta_0$ , size of Fireflies (population)  $P$ , and a maximum of generation number  $T$ .
- 3: Generate the initial population  $x_i$  randomly,  $i = \{1, 2, \dots, P\}$
- 4: Based on the intensity of the fireflies' light, compute the fitness of the initial population.
- 5: for  $t$  in range(1,  $T$ )
- 6: for  $i$  in range(1,  $P$ )
- 7: for  $j$  in range(1,  $P$ )
- 8: determine light intensity  $I$ , (Eq. 2).
- 9: compute the distance of acceleration between  $x_i$  and  $x_j$ , (Eq. 4).
- 10: calculate acceleration (Eq. 9, Eq. 7, Eq. 10, and Eq. 12) respectively.
- 11: if ( $I(i) < I(j)$ )
- 12: Firefly  $i$  is moved towards Firefly  $j$  (Eq. 17).
- 13: Else
- 14: Firefly  $i$  is moved randomly towards Firefly  $j$  (Eq. 18).
- 15: end
- 16: calculate the new solutions and update the light intensity value
- 17: end
- 18: end
- 19: end
- 20: descending order for the fireflies based on their light intensity.

A Firefly algorithm (FA) operator mutates the light intensity attraction step of each particle in the proposed method. That means each particle is attracted to the best position in the entire group at random. The modified attractiveness step of the FAGSA algorithm performs local searches in different regions. The FAGSA feature selection stage's main goal is to reduce the problem's features before supervising neural network classification. The FAGSA algorithm has the distinction of being able to be used to solve optimization problems using the methods of firefly flash behavior as a promising wrapper algorithm among all the ones used.

To demonstrate FAGSA's efficiency, the following remarks are made: The quality of the results and good solutions (fitness) are taken into account when updating the FAGSA algorithm. Particles near good solutions attract others that explore the search space. When all the particles get close to the good solutions, they move slowly. In this case, the acceleration property of the GSA algorithm is used to combine with the FA algorithm to utilize the best solution and increase the acceleration to move into the best firefly brightness. The best FAGSA solutions are stored in memory where they can be extracted and accessed at any time. Therefore, each particle can be observed and attracted to the optimal solution.

## 2. Experimental results and discussion

The new hybrid (FAGSA), FA, and GSA models are implemented in MATLAB R2015a individually. FAGSA's performance is evaluated using twenty-three standard benchmark functions used for testing the performance of FA and GSA algorithms. Tables 1, 2, 3, and 4 lists these benchmark functions, their dimensions, their search space ranges, and their optimum fitness values. A detailed description of these functions is available in [27].

**Table 1: Benchmark Functions (unimodal function)**

Function		Dim <sup>a</sup>	Range	Optimum
Sphere	$F_1(x) = \sum_{i=1}^n x_i^2$	30	[-100, 100]	<b>0</b>
Schwefel 2.22	$F_2(x) = \sum_{i=1}^n  x_i  + \prod_{i=1}^n  x_i $	30	[-10, 10]	<b>0.00032</b>
Schwefel 1.2	$F_3(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	30	[-100, 100]	<b>0</b>
Schwefel 2.21	$F_4(x) = \max_i  x_i , 1 \leq i \leq n$	30	[-100, 100]	<b>0</b>
Rosen Brock	$F_5(x) = \sum_{i=1}^n [100(x_{i+1} - x_i^2)^2] - (x_i - 1)^2$	30	[-30, 30]	<b>26.70000</b>
Step	$F_6(x) = \sum_{i=1}^n ([x_i - 0.5])^2$	30	[-100,100]	<b>3</b>
Quartic	$F_7(x) = \sum_{i=1}^n ix_i^4 + \mathbf{rand}[0, 1)$	30	[-1.28, 1.28]	<b>0.01130</b>

a. Indicates the number of variables (dimension of the functions)

Table 2- Benchmark Functions (multimodal function)

Function		Dim <sup>a</sup>	Range	Optimum
Schwefel	$F_8(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	30	[-500, 500]	(3,320)
Rastrigin	$F_9(x) = \sum_{i=1}^n  x_i^2 - 10 \cos(2\pi x_i) + 10 $	30	[-5.12, 5.12]	45.5
Ackley	$F_{10}(x) = -20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$	30	[-32, 32]	<b>0</b>
Griewank	$F_{11}(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	30	[-600, 600]	<b>0.00740</b>
Penalized	$F_{12}(x) = \frac{\pi}{n} \{10 \sin^2(\pi y_1) + \sum_{i=1}^n (y_i - 1)^2 [1 + \sin^2(3\pi x_i) + \sum_{i=1}^n u(x_i, 10, 100, 4)]\}$	30	[-50,50]	<b>0</b>
Penalized2	$F_{13}(x) = 0.1 \{ \sin^2(3\pi x_1) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] \} + \sum_{i=1}^n u(x_i, 10, 100, 4)$	30	[-50,50]	<b>0</b>

a. Indicates the number of variables (dimension of the functions)

Table 3- Benchmark Functions (fixed-dimensional multimodal function)

Function		Dim <sup>a</sup>	Range	Optimum
Foxholes	$F_{14}(x) = (\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6})^{-1}$	2	[-65.536, 65.536]	2.77
Kowalik	$F_{15}(x) = \sum_{i=1}^{11} [a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}]^2$	4	[-5, 5]	0.00031
Six Hump Camel	$F_{16}(x) = 4x_1^2 + 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[-5, 5]	<b>(1.03)</b>
Branin	$F_{17}(x) = \sum_{i=1}^n (x_i - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6)^2 + 10 (1 - \frac{1}{8\pi}) \cos x_1 + 10$	2	[-5, 5]	<b>0.398</b>
Goldstein-Price	$F_{18}(x) = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] + [30 + (2x_1 - 3x_2)^2 x(18 - 32x_1 + 12x_2^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	[-2, 2]	<b>3</b>



Hartman 3	$F_{19}(x) = \sum_{i=1}^4 c_i \exp(-\sum_{j=1}^3 a_{ij}(x_j - p_{ij})^2)$	4	[1, 3]	(3.86)
Hartman 6	$F_{20}(x) = \sum_{i=1}^4 c_i \exp(-\sum_{j=1}^6 a_{ij}(x_j - p_{ij})^2)$	6	[0, 1]	(3.32)
Shekel 5	$F_{21}(x) = \sum_{i=1}^5 [(X - a_i)(X - a_i)^T]^{-1}$	4	[0, 10]	(8.45)
Shekel 7	$F_{22}(x) = \sum_{i=1}^7 [(X - a_i)(X - a_i)^T]^{-1}$	4	[0, 10]	(10.4)
Shekel 10	$F_{23}(x) = \sum_{i=1}^{10} [(X - a_i)(X - a_i)^T]^{-1}$	4	[0, 10]	(10.5)
a. Indicates the number of variables (dimension of the functions)				

In this paper, our objective is minimization optimization. Therefore, many parameters should be initialized in the algorithms FA, GSA, and FAGSA. For FA was performed with the following parameters: absorption coefficient (gamma=0.01), randomness reduction factor (theta=10<sup>-5</sup>/maxiter) ≈ 0.97, attractiveness constant/light amplitude (beta=1.0), maximum iteration (maxiter)=1000, and stop when criteria equal maxiter. For GSA and FAGSA, the following settings were used: population size=50, G0=100, α=20, and maximum iteration = 1000, and stopping when criteria equal maxiter.

**Table 4 :** Results of twenty-three Benchmark Functions

Fun	Standard of FA		Standard of GSA		FAGSA	
	Fitness value	Mean value	Fitness value	Mean value	Fitness value	Mean value
F1	<b>2.17E-22</b>	-5.59E-13	9.18E+02	9.18E+02	4.45E-17	<b>-1.68E-10</b>
F2	1.64E+02	<b>-7.09E-01</b>	9.99E+00	9.99E+00	<b>3.17E-04</b>	-2.19E-06
F3	1.17E+05	<b>-4.91E-01</b>	6.07E+02	6.07E+02	<b>2.18E-12</b>	5.80E-09
F4	8.58E+01	3.41E+00	2.07E+01	2.07E+01	<b>4.19E-09</b>	<b>-2.20E-11</b>
F5	9.25E+07	<b>-2.26E+00</b>	1.09E+04	1.09E+04	<b>2.67E+01</b>	5.47E-02
F6	5.94E+04	<b>-7.46E+00</b>	1.68E+03	1.68E+03	<b>3.00E+00</b>	3.78E-02
F7	6.29E+00	1.06E-01	3.81E-01	1.36E+01	<b>1.13E-02</b>	<b>-5.98E-03</b>
F8	-2.47E+03	1.05E+02	<b>-2.66E+03</b>	<b>-4.34E+01</b>	-3.32E+03	-3.80E+02
F9	1.72E+02	1.88E-01	<b>4.10E+01</b>	4.10E+01	4.55E+01	<b>1.56E-01</b>
F10	2.08E+01	<b>-3.12E+00</b>	1.16E+01	1.50E+01	<b>1.73E-09</b>	-3.50E-11
F11	5.82E+02	<b>-5.59E+01</b>	1.51E+00	1.51E+00	<b>7.40E-03</b>	2.53E-01
F12	5.69E+08	4.45E+00	6.56E+00	6.56E+00	<b>1.82E-19</b>	<b>-1.00E+00</b>
F13	1.16E+09	<b>6.56E-02</b>	1.12E+04	1.12E+04	<b>5.99E-17</b>	1.00E+00
F14	9.98E-01	<b>-3.20E+01</b>	<b>1.05E+00</b>	4.63E+02	2.77E+00	-3.14E+01
F15	<b>4.11E-03</b>	1.57E+00	6.11E-04	<b>6.11E-04</b>	3.07E-04	1.61E-01
F16	-9.07E-01	<b>2.47E-01</b>	<b>-1.03E+00</b>	<b>-1.03E+00</b>	<b>-1.03E+00</b>	3.11E-01
F17	5.60E-01	2.87E+00	<b>3.98E-01</b>	<b>3.98E-01</b>	<b>3.98E-01</b>	2.71E+00
F18	5.28E+01	-4.29E-01	<b>3.00E+00</b>	3.00E+00	<b>3.00E+00</b>	<b>-5.00E-01</b>
F19	-3.06E+00	6.78E-01	-3.85E+00	<b>-1.09E+00</b>	-3.86E+00	5.08E-01
F20	-1.45E+00	4.19E-01	<b>-3.32E+00</b>	<b>-3.32E+00</b>	<b>-3.32E+00</b>	3.44E-01
F21	-1.94E+00	4.01E+00	-2.68E+00	<b>-2.68E+00</b>	<b>-8.45E+00</b>	3.98E+00
F22	-3.12E+00	4.22E+00	-5.26E+00	<b>-4.45E-01</b>	<b>-1.04E+01</b>	4.00E+00
F23	-2.56E+00	4.92E+00	<b>-1.05E+01</b>	<b>-1.05E+01</b>	<b>-1.05E+01</b>	4.00E+00

Table 4 shows the experimental results. According to dominated and non-dominated rules, the results are an average of 50 runs, with the *best results* highlighted in *bold* type.

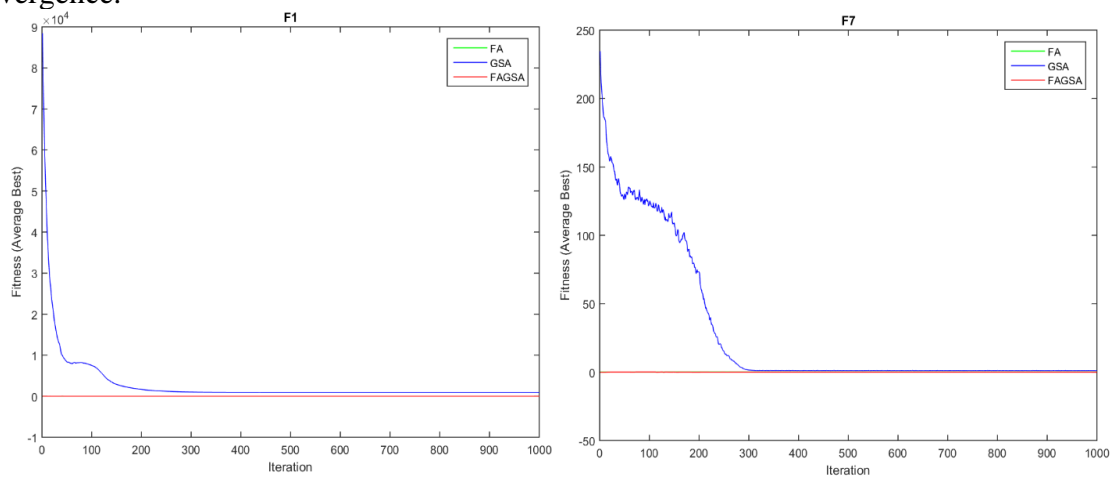
FA dominated FAGSA in terms of the *best fitness* on two functions: (F1 and F15) and in terms of the *best mean* on nine functions: (F2, F3, F5, F6, F10, F11, F13, F14, and F16).

GSA dominated FAGSA in terms of the *best fitness* values on eight functions only: F8, F9, F14, F16-F18, F20, and F23, and in terms of the *best mean* values on nine functions only: F8, F15-F17, and F19-F23.

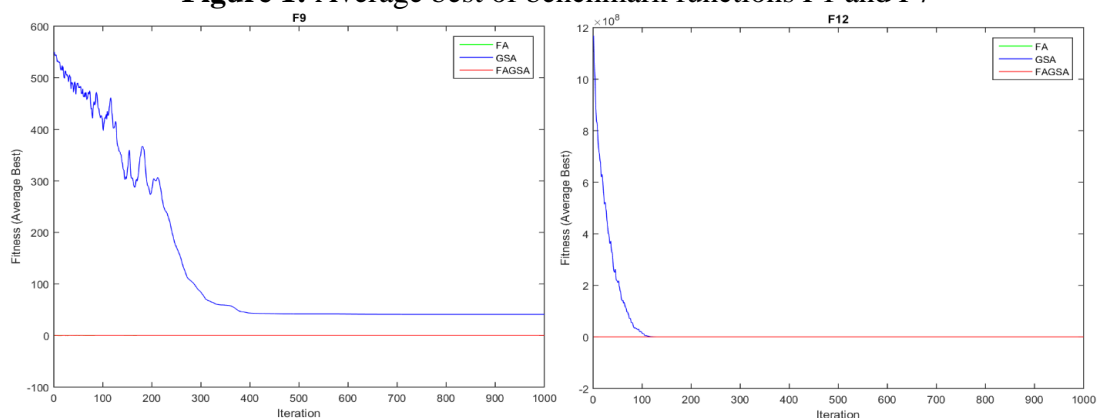
That is, for the best fitness values of fifty iterations on twenty-three different fitness functions, the FAGSA performs best on seventeen functions, FA performs the best on two functions, GSA performs the best on four functions, and saddle point performs the best on the remaining four functions (F16-F18, F20, and F23).

As a result, the FAGSA performs nearly twice as many functions as the FA and the GSA. FAGSA achieves global minima in all benchmark functions except F8, F14, F15, and F19 for the best fitness of 50 runs. F19 is a noisy problem, and all algorithms exhibit similar characteristics.

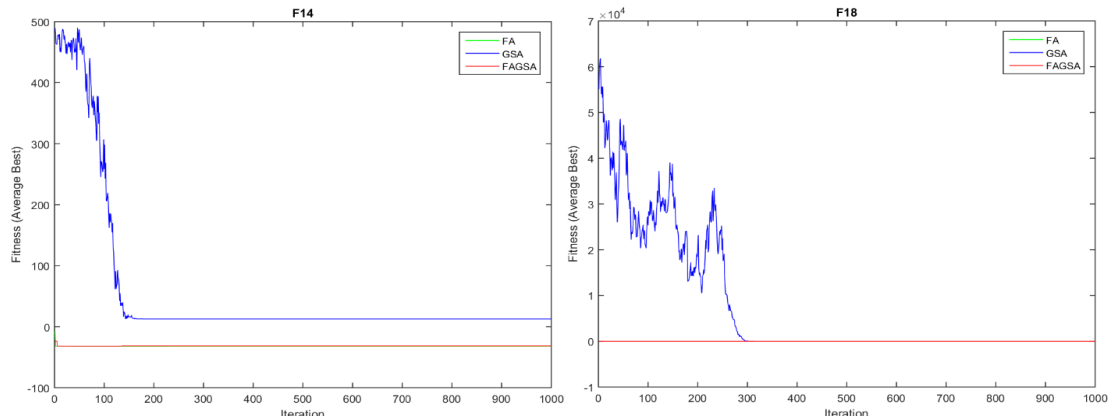
FAGSA, like FA and GSA, can find global minima in 50 runs for the functions F8, F22, and F23. FAGSA performs better on functions with wide domains because the functions F22 and F23 all have narrow domains. FAGSA only finds global minima for high-dimensional functions (F1 to F13) on F8 and F12, This implies that FAGSA performs well on high-dimensional functions. Figure 1, Figure 2, and the following Figure 3 for the selected functions show the new algorithm FAGSA outperforms the standard FA and GSA in terms of the rate of convergence.



**Figure 1:** Average best of benchmark functions F1 and F7



**Figure 2:** Average best of benchmark functions F9 and F12



**Figure 3:** Average best of benchmark functions F14 and F18

### 3. Conclusion

This paper proposes a new numerical optimization method based on the hybridization of the two methods, namely FA and GSA. The proposed method gives, in most cases, better results than the original ones. The core concept of the new approach is to hybridize the facility exploitation of the Firefly algorithm and the facility exploration of the capabilities of the Gravitational Search Algorithm. Twenty-three benchmark functions are used to validate the new algorithm FAGSA's performance in comparison to standard FA and GSA. According to the results, the new algorithm FAGSA performs better than both the FA algorithm and the GSA algorithm in most minimization function tasks. The results also prove that the new algorithm FAGSA has a faster convergence speed than FA and GSA.

The main findings of this study can be summarized as follows:

- The new numerical hybrid optimization technique FAGSA based on the FA and the GSA, is proposed in this paper to address energy consumption forecasting in residential households.
- The performance of the proposed algorithm is compared with other well-known optimization techniques such as FA, GSA, and PSO using the objective function.
- The computation time is almost the same for all algorithms analyzed in this study, so the complexity of the proposed algorithm remains comparable to the basic algorithms and is used as a benchmark for evaluating performance.

Hence, it can be inferred that the FAGSA outperforms the other optimization techniques (FA, GSA, and PSO). Therefore, it can be successfully applied for energy consumption forecasting.

### 4. Authors' declaration:

- Conflicts of Interest: None.
- We hereby confirm that all the Figures and Tables in the manuscript are mine. Besides, the Figures and images, which are not mine, have been given permission for re-publication and attached with the manuscript.
- The author has signed an animal welfare statement.
- Authors sign on ethical consideration's approval
- Ethical Clearance: The project was approved by the local ethical committee at Cairo University.

## 5. Authors' Contribution:

A. Mohamed, Z. Hegazy, Naglaa, and O. Eyman contributed to the design and implementation of the research, the analysis of the results, and the writing of the manuscript (%50 Mohamed, %20 Hegazy, %15 Naglaa, and %15 Eyman). All authors discussed the results and contributed to the final manuscript.

## 6. Acknowledgements

The authors would like to thank the Iraqi Journal of Science (IJS), University of Baghdad – College of Science, for their support in making this study a success.

## References

- [1] V. Dasari, M. S. Im, and L. Beshaj, "Solving machine learning optimization problems using quantum computers," no. April, p. 16, 2020, doi: 10.1117/12.2565038.
- [2] A. Bouaouda and Y. Sayouti, "Hybrid Meta-Heuristic Algorithms for Optimal Sizing of Hybrid Renewable Energy System: A Review of the State-of-the-Art," *Arch. Comput. Methods Eng.*, vol. 1, p. 3, 2022, doi: 10.1007/s11831-022-09730-x.
- [3] S. A. Yasear and K. R. Ku-Mahamud, "Taxonomy of memory usage in swarm intelligence-based metaheuristics," *Baghdad Sci. J.*, vol. 16, no. 2, pp. 445–452, 2019, doi: 10.21123/bsj.2019.16.2(SI)0445.
- [4] X. S. Yang, "A new metaheuristic Bat-inspired Algorithm," *Stud. Comput. Intell.*, vol. 284, pp. 65–74, 2010, doi: 10.1007/978-3-642-12538-6\_6.
- [5] X. S. Yang and S. Deb, "Cuckoo search via Lévy flights," *2009 World Congr. Nat. Biol. Inspired Comput. NABIC 2009 - Proc.*, pp. 210–214, 2009, doi: 10.1109/NABIC.2009.5393690.
- [6] K. S. Tang, K. F. Man, S. Kwong, and Q. He, "Genetic algorithms and their applications," *IEEE Signal Process. Mag.*, vol. 13, no. 6, pp. 22–37, 1996, doi: 10.1109/79.543973.
- [7] X. S. Yang, "Firefly algorithms for multimodal optimization," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5792 LNCS, pp. 169–178, 2009, doi: 10.1007/978-3-642-04944-6\_14.
- [8] X. S. Yang, "Firefly algorithm, stochastic test functions and design optimization," *Int. J. Bio-Inspired Comput.*, vol. 2, no. 2, pp. 78–84, 2010, doi: 10.1504/IJBIC.2010.032124.
- [9] S. Rainer and P. Kenneth, "Differential Evolution: A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," *J. Glob. Optim.*, vol. 11, no. 4, p. 341, 1997, [Online]. Available: <http://ezlibproxy1.ntu.edu.sg/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=e do&AN=ejs644912&site=eds-live&scope=site>
- [10] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "GSA: A Gravitational Search Algorithm," *Inf. Sci. (Ny)*, vol. 179, no. 13, pp. 2232–2248, 2009, doi: 10.1016/j.ins.2009.03.004.
- [11] M. Dorigo, V. Maniezzo, and A. Coloni, "Dorigo-Maniezzo-Coloni\_the-Ant-System-Optimization-By-a-Colony-of-Cooperating-Agents," *IEEE Trans. Syst. Man, Cybern. B*, vol. 26, no. 1, pp. 1–26, 1999, [Online]. Available: <papers://82ac23f7-2eaf-4339-a5e1-4600c19d7f01/Paper/p2331>
- [12] A. Slowik, "Particle Swarm Optimization," *Proc. IEEE Int. Conf. Evol. Comput.*, pp. 1942–1948, 1995, doi: 10.1007/978-3-319-46173-1\_2.
- [13] Y. Shi and R. Eberhart, "A Modified Particle Swarm Optimizer," *Proc. IEEE Int. Conf. Evol. Comput.*, no. June, pp. 69–73, 1998, doi: 10.1109/ICEC.1998.699146.
- [14] KARABOGA, "An idea based on honey bee swarm for numerical optimization," *Mater. Japan*, vol. 44, no. 1, pp. 24–31, 2005, doi: 10.2320/materia.44.24.
- [15] N. Kamel and T. El-Omari, "Sea Lion Optimization Algorithm for Solving the Maximum Flow Problem," *IJCSNS Int. J. Comput. Sci. Netw. Secur.*, vol. 20, no. 8, p. 30, 2020, doi: 10.22937/IJCSNS.2020.20.08.5.
- [16] M. S. Jørgensen, U. F. Larsen, K. W. Jacobsen, and B. Hammer, "Exploration versus Exploitation in Global Atomistic Structure Optimization," *J. Phys. Chem. A*, vol. 122, no. 5, pp. 1504–1509, 2018, doi: 10.1021/acs.jpca.8b00160.
- [17] J. Mcdermott, "When and Why Metaheuristics Researchers can Ignore 'No Free Lunch'"

- Theorems,” *SN Comput. Sci.*, vol. 1, p. 60, 2020, doi: 10.1007/s42979-020-0063-3.
- [18] P. Kora and K. S. Rama Krishna, “Hybrid firefly and Particle Swarm Optimization algorithm for the detection of Bundle Branch Block,” *Int. J. Cardiovasc. Acad.*, vol. 2, no. 1, pp. 44–48, 2016, doi: 10.1016/j.ijcac.2015.12.001.
- [19] A. Khan, H. Hizam, N. I. Abdul-Wahab, and M. L. Othman, “Solution of optimal power flow using non-dominated sorting multi objective based hybrid firefly and particle swarm optimization algorithm,” *Energies*, vol. 13, no. 6, 2020, doi: 10.3390/en13164265.
- [20] Y. D. Lina Zhang, Liqiang Liu, Xin-She Yang, “A Novel Hybrid Firefly Algorithm for Global Optimization \_ Enhanced Reader.pdf,” *IEEE 4th Int. Conf. Comput. Commun. Syst. ICCCS*, pp. 164–168, 2019, doi: DOI: 10.1109/CCOMS.2019.8821670.
- [21] M. A. Tawhid and A. F. Ali, “An effective hybrid firefly algorithm with the cuckoo search for engineering optimization problems,” *Math. Found. Comput.*, vol. 1, no. 4, pp. 349–368, 2018, doi: 10.3934/mfc.2018017.
- [22] O. Avalos, “GSA for machine learning problems: A comprehensive overview,” *Appl. Math. Model.*, vol. 92, pp. 261–280, 2021, doi: 10.1016/j.apm.2020.11.013.
- [23] M. Khajehzadeh, M. R. Taha, and M. Eslami, “A new hybrid firefly algorithm for foundation optimization,” *Natl. Acad. Sci. Lett.*, vol. 36, no. 3, pp. 279–288, 2013, doi: 10.1007/s40009-013-0129-z.
- [24] S. Sennan *et al.*, “Energy efficient optimal parent selection based routing protocol for Internet of Things using firefly optimization algorithm,” 2020, doi: 10.1002/ett.4171.
- [25] H. N. K. Al-Behadili, “Improved firefly algorithm with variable neighborhood search for data clustering,” *Baghdad Sci. J.*, vol. 19, no. 2, pp. 409–421, 2022, doi: 10.21123/BSJ.2022.19.2.0409.
- [26] Isaac Newton, *In experimental philosophy particular propositions are inferred from the phenomena and afterwards rendered general by induction*, 3rd ed. Andrew Motte’s English translation published, 1729, vol. 2.
- [27] W. Zhao, L. Wang, and Z. Zhang, “Atom search optimization and its application to solve a hydrogeologic parameter estimation problem,” *Knowledge-Based Syst.*, vol. 163, pp. 283–304, 2019, doi: 10.1016/j.knosys.2018.08.030.