

## Article

# Gaussian Mutation Specular Reflection Learning with Local Escaping Operator Based Artificial Electric Field Algorithm and Its Engineering Application

Oluwatayomi Rereloluwa Adegbeye <sup>1</sup>  and Ezgi Deniz Ülker <sup>2,\*</sup> <sup>1</sup> Computer Engineering Department, European University of Lefke, Mersin 10, Turkey; 194020@std.eul.edu.tr<sup>2</sup> Software Engineering Department, European University of Lefke, Mersin 10, Turkey

\* Correspondence: eulker@eul.edu.tr

**Abstract:** During the contribution of a metaheuristic algorithm for solving complex problems, one of the major challenges is to obtain the one that provides a well-balanced exploration and exploitation. Among the possible solutions to overcome this issue is to combine the strengths of the different methods. In this study, one of the recently developed metaheuristic algorithms, artificial electric field algorithm (AEFA), has been used, to improve its converge speed and the ability to avoid the local optimum points of the given problems. To address these issues, Gaussian mutation specular reflection learning (GS) and local escaping operator (LEO) have been added to the essential steps on AEFA and called GSLEO-AEFA. In order to observe the effect of the applied features, 23 benchmark functions as well as engineering and real-world application problems were tested and compared with the other algorithms. Friedman and Wilcoxon rank-sum statistical tests, and complexity analyses were also conducted to measure the performance of GSLEO-AEFA. The overall effectiveness of the algorithm among the compared algorithms obtained in between 84.62–92.31%. According to the achieved results, it can be seen that GSLEO-AEFA has precise optimization accuracy even in changing dimensions, especially in engineering optimization problems.

**Keywords:** artificial electric field algorithm; local escaping operator; Gaussian mutation; specular reflection learning



**Citation:** Adegbeye, O.R.; Deniz Ülker, E. Gaussian Mutation Specular Reflection Learning with Local Escaping Operator Based Artificial Electric Field Algorithm and Its Engineering Application. *Appl. Sci.* **2023**, *13*, 4157. <https://doi.org/10.3390/app13074157>

Academic Editors: Antonio J. Nebro and José Manuel García-Nieto

Received: 2 March 2023

Revised: 19 March 2023

Accepted: 23 March 2023

Published: 24 March 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The development of optimization techniques is crucial since there are many optimization issues that need to be resolved in the real world. The majority of these techniques depend on being able to find derivatives of the involved functions. However, for various reasons, the derivatives might be occasionally challenging to get. Those problems include electromagnetic problems which were difficult to tackle because of limitations in computational sources [1]. However due to the advancement of computers many different optimization methods are applied to solve these kind of scattering problems [2,3]. Metaheuristic algorithms which are a crucial component of derivative-free approaches are becoming more and more popular due to their effective searching capabilities. Metaheuristic algorithms can be categorized into groups based on what they imitate or mimic. Swarm intelligence mimicking algorithms are gaining popularity due to a variety of features including powerful searching capabilities, straightforward implementation, minimal parameters, and the capacity to avoid becoming stuck in a sub-optimal position [4]. The search agents collaborate and share information with one another in these algorithms, ensuring that the search space's information is effectively used. This allows the entire swarm to advance toward more promising areas of the search space. Numerous swarm intelligence techniques have been presented in several of the literatures and used to solve practical issues during the last few decades [5–10]. Physics law mimicking algorithms imitate the physical principles that control how agents interact with one another and their search environment.

These principles include the principle of gravity, light, and refraction [11]. They are being implemented in several areas to solve engineering challenges [12–15].

Evolution theory mimicking algorithms through various combination and permutations of populations obtain the best solution. The main benefit of this method is that the individuals with the best solutions combine to create the next generation [16–19]. The exploitation and exploration patterns of behavior are basically the patterns the metaheuristic algorithms will follow as it advances towards the optimal solution. The discovery of new search space areas is called exploration. Exploitation is the procedure of locating and investigating the most promising areas in the hunt for potentially superior solutions [20]. Any metaheuristic algorithms share a potential problem in trying to maintain an adequate balance between these two patterns since they contradict each other [21].

In 2019, Anita and Yadav suggested the artificial electric field algorithm (AEFA) [22]. The leadership influence of the lead particle, in other words, the particle with the strongest charge and the behavior of the other particles as they are drawn along the search area by the lead particle, serve as the foundation for the search mechanism of the AEFA. This approach ensures that the algorithm is able to exploit promising areas for the best solution consistently [15]. AEFA has earned a reputation and has been effectively applied to several sectors in recent years due its various benefits, including ease of implementation, low computing cost, and great search efficiency. A few of the areas in which AEFA has proved a useful optimization algorithm are tumor detection [23], multi-objective problem [24], parameter extraction for photovoltaic model [25], assembly line balancing optimization problem [26], and feature selection [27]. Despite AEFA being applicable in many areas, prior research has revealed that it has significant shortcomings such as premature convergence, exploration and exploitation imbalance, and a propensity to become stuck at sub-optima [15,28,29], which are also present in other metaheuristic algorithms. However, the attraction force mechanism of the AEFA has a significant impact on population diversification. In the typical AEFA, a dominating particle directs the search process. This causes the population to gather around a single particle. As a result, this particle's information is crucial for the population, and when it results in a less-than-ideal solution, it lowers the search efficiency of the algorithm.

The absence of techniques for population mutation, the propensity to enter sub-optimal and delayed convergence are some of the issues that restrict the practical uses of AEFA, despite the fact that it is an effective optimization technique. To address the shortcomings of AEFA, researchers have produced a number of variants in recent years. To improve the search efficiency and stability of AEFA, the artificial electric field algorithm with pattern search algorithm (AEFA-PS) was suggested for open switch network problem. It was shown to be capable of getting objective function fitness values of greater quality [30]. The improved artificial electric field algorithm (IAEFA) developed a unique method for calculating the electrostatic force that solved the limitations of AEFA such as early convergence and low search performance. The performance of IAEFA was assessed using 18 different test functions and it was discovered to be a more effective and efficient at solving the optimization problems [31]. Although AEFA has limited global search capabilities, it offers excellent local search capabilities. In order to overcome this, the adaptive search sine-cosine algorithm with artificial electric field algorithm (SC-AEFA) technique was created to dynamically balance the global and local search abilities of the algorithm. Experiments demonstrated the greater performance was achieved [15]. In order to overcome the problem of entering the sub-optimality in the original AEFA, the Moth Levy adopted artificial electric field algorithm (ML-AEFA) was proposed. In comparison to other algorithms, experiments revealed superior performance [32]. By instantly introducing the fittest particles into the population for the following generation and keeping the best particles from one iteration to the next, the elitism strategy boosts the potential of AEFA for optimization. Studies demonstrate the higher performance of AEFA [33].

The primary driving force for this study is to solve the aforementioned issues. Our goal is to successfully strike a balance between the capacity for exploration and exploitation

and the enhancement of population variety. The suggested AEFA is based on Gaussian mutation specular reflection (GS) and local escaping operator (LEO) called GSLEO-AEFA. Firstly, the convergence efficiency of an algorithm is impacted by the low diversity and the use of starting agents that are randomly generated. The specular reflection learning approach is applied in this study to produce high-quality starting populations. Gaussian mutation is introduced for the mutation of the new population to further increase the variety of the starting population. Additionally, the local escape operator mechanism is added to the algorithm to update particle positions and prevent the population from settling into sub-optimal, thereby enhancing the capabilities for global search. It is also designed to advance the population toward the global optimum and achieve equilibrium between exploration and exploitation.

During the forming of the GSLEO-AEFA, no modifications have been applied on the combined methods except the opposition-based learning (OBL). In this paper, we proposed an enhanced form of OBL called Gaussian mutation-based specular reflection (GS) and applied it to the standard AEFA algorithm. OBL is a technique that generates diverse candidate solutions in multimodal problems by having each solution learn from its opposite candidate [34]. It is a principle that inspired from the development of specular reflection. Specular reflection models the reflection of light in physics [35]. Our approach involves introducing the mutated form of specular reflection using Gaussian mutation only during the population initialization stage to enhance the diversity. This differs from the OBL introduced by Alkayem et al. [36] which is applied at initialization and later during iteration, and the quasi-based opposition learning introduced by Alkayem et al. [37] which aims to improve convergence speed and escape local optima during both initialization and iteration.

The main contributions of this paper are:

1. We introduce a GSLEO-AEFA algorithm to improve the performance and balance in between exploitation and exploration phases of the original AEFA.
2. We introduce a Gaussian mutation with specular learning strategy to enhance population diversity and as a new strategy for population initialization.
3. We introduce a local escaping operator as a method for avoiding stagnation in local optimal.
4. We use real world engineering problems to evaluate GSLEO-AEFA.

The remainder of the paper is organized as follows. Section 2 contains the detailed explanations of the artificial electrical field optimization algorithm (AEFA) and the proposed Gaussian mutation specular refraction learning with local escaping operator based AEFA (GSLEO-AEFA). Section 3 shows the experimental results and discussions of GSLEO-AEFA with its counterpart algorithms using 23 benchmark functions and engineering problems. Some necessary analyses such as convergence trajectory analysis, complexity analysis, diversity analysis and statistical analyses are implemented and discussed here. Lastly, in Section 4 the conclusion of the study is given.

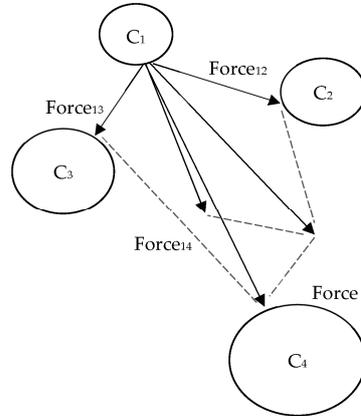
## 2. Methodology

### 2.1. Artificial Electric Field Algorithm (AEFA)

The artificial electric field algorithm (AEFA) imitates how charged particles flow in an electrostatic field. The repulsive force between charged particles is ignored by AEFA, and only the attractive force between particles is considered. This enables surrounding low-charge particles to be drawn by high-charge particles in the electric field.

Each charged particle in the search area represents a potential solution. The charge associated with a particle determines the quality of the solution. The solution is closer to the ideal solution as the charge increases. Figure 1 illustrates the motion of the charged particles where the circles represent the several particles as they are located across the electric field and the circle sizes represent the quantity values of the charges that the particles carry. Particle  $C_1$  experiences a force named Force in Figure 1 and it accelerates in the direction of the other three charged particles as a result of being attracted towards them. As can be seen,

$C_4$  has the highest charge and it strongly attract  $C_1$ . As a result, the direction of the force acting on  $C_1$  is more closely drawn towards the direction of  $C_4$ . Particles in the search area with low charges would therefore gravitate towards those with high charges throughout the AEFA iteration phase, enabling the algorithm to reach to the best outcome [22,38].



**Figure 1.** Illustration of the forces exerted on charges.

Equations (1) and (2) explains the theoretical formulation of the AEFA [39].

$$Vel_i^{t+1} = r \times Vel_i^t + accl_i^t \tag{1}$$

$$P_i^{t+1} = P_i^t + Vel_i^t \tag{2}$$

The  $i$ th charged particle’s velocity is given as  $Vel_i^{t+1}$  and the location of that particle is represented with  $P_i^{t+1}$  using Equations (1) and (2). The velocity and current position can be updated at iteration  $t + 1$ ;  $r$  denotes random number in the range  $[0, 1]$ . The acceleration of particle at the  $t$  iteration is denoted by  $accl_i^t$ .

The formula of acceleration when a particle is exposed to an electric fields’ force is defined in Equation (3) as follows by Newton’s second law,

$$accl_i^t = \frac{C_i^t \times EF_i^t}{m_i^t} \tag{3}$$

where  $m_i^t$  denote mass of a particle and  $C_i^t$  represent the charge of a particle. The electric field intensity of the  $i$ th particle at iteration  $t$  is expressed as,

$$EF_i^t = \frac{Force_i^t}{C_i^t} \tag{4}$$

The charged particle’s resulting force in the search space is given as,

$$Force_i^t = \sum_{j=1, j \neq i}^N r \times Force_{ij}^t \tag{5}$$

According to the Coulomb’s force, the resulting force exacted on particle  $i$  by  $j$  is computed as follows.

$$Force_{ij}^t = K^t \times \frac{C_i^t \times C_j^t \times (P_{best}^t - P_i^t)}{R_{ij}^t + \epsilon} \tag{6}$$

$$R_{ij} = \|P_i(t), P_j(t)\|_2 \tag{7}$$

$$K^t = K_0 \times \exp\left(-\alpha \times \frac{\text{iteration}}{\text{Max}_{\text{iteration}}}\right) \tag{8}$$

In Equation (7),  $R_{ij}$  denote the distance between two agents  $i$  and  $j$ ,  $P_{\text{best}}^t$  is the particle with ideal solution at the iteration  $i$ , and  $\epsilon$  denote a very tiny positive value. The Coulomb constant  $K_0$  is initially set at 500 and drops by a factor of 30 which is denoted as  $\alpha$  in Equation (8) at every iteration. This progressive decline enables the algorithm to start with a global search and switch to a local search later on.

Each particles charge  $C_i^t$  at a given iteration as given as follows,

$$C_i^t = \frac{cq_i^t}{\sum_{i=1}^N cq_i^t} \tag{9}$$

$$cq_i^t = \exp\left(\frac{\text{fit}_i^t - \text{worst}(t)}{\text{best}(t) - \text{worst}(t)}\right) \tag{10}$$

In Equation (9), the total number of particles which is represented by  $N$ , the normalization of  $cq_i^t$  is denoted by  $C_i^t$ , where  $cq_i^t$  is the charge of the  $i$ th particle. In Equation (10),  $\text{best}(t)$  and  $\text{worst}(t)$  denote the ideal and worst fitness values at the  $t$ th iteration, respectively.  $\text{fit}_i^t$  is the fitness value of the  $i$ th particle. Using a greedy method, the algorithm updates the positions particles at each iteration.

$$P_i^{t+1} = \begin{cases} P_i^t & \text{when fit}(P_i^t) < \text{fit}(P_i^{t+1}) \\ P_i^{t+1} & \text{when fit}(P_i^t) \geq \text{fit}(P_i^{t+1}) \end{cases} \tag{11}$$

The position of the individual in the following generation is determined by its fitness, according to Equation (11). The individual’s position is unaffected, if it has less fitness value than the fitness value of newly generated individual. If not, its position is re-adjusted. The pseudocode of AEFA can be seen in the following Algorithm 1.

---

**Algorithm 1** Artificial Electric Field Algorithm (AEFA)

---

```

Initialize the population of size  $N$  within search range  $[P_{\min}, P_{\max}]$ 
Set the Initial  $Vel_i^t$  to 0
Calculate fitness value for all particles
Set iteration  $i \leftarrow 1$ 
while stopping requirement is not met do
    Compute  $K^t$ ,  $\text{best}(t)$  and  $\text{worst}(t)$ 
    for  $i \leftarrow 1$  to  $N$  do
        Calculate fitness values
        Calculate the total force in each direction Equation (5)
        Calculate acceleration Equation (3)
         $Vel_i^{t+1} \leftarrow r \times Vel_i^t + accl_i^t$ 
         $P_i^{t+1} \leftarrow P_i^t + Vel_i^t$ 
    end for
end while
    
```

---

2.2. Proposed GSLEO-AEFA

2.2.1. Population Initialization Using Gaussian Mutation Specular Reflection Learning (GS)

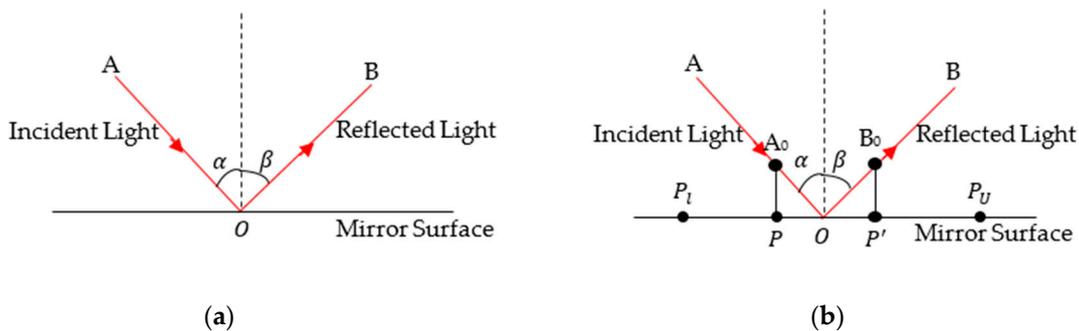
The variety of the starting agents has a major influence on the convergence rate and solution precision of metaheuristic algorithms [40]. However, AEFA frequently uses random initialization to generate the starting population. This initialization strategy lacks prior information of the search space, which might impact the updating method of the search agents. The population has the possibility of moving away from the ideal solution if the optimal solution is situated in the opposite location from where the randomly produced

individuals are positioned. Solutions generated by the specular reflection learning (SRL) technique has been shown to outperform those produced only by random approaches [41]. This study introduces the application of specular reflection learning for the initialization of population and will produce an opposing population of the initial population. Afterwards, the Gaussian mutation (GM) process will be applied to the newly generated population to improve population variety from the generated agents. It is decided which individuals should be retained by comparing their fitness values with that of the original individuals.

Due to its superior performance, opposition-based learning (OBL) is frequently employed to enhance metaheuristic algorithms [42]. OBL uses simultaneous examination of potential solution and its opposing solution to promote rapid convergence. Within the range  $[lb, ub]$ , the opposing value of  $p$  is defined as follows:

$$p^* = lb + ub - p \tag{12}$$

Zhang introduced the specular reflection learning (SRL) which was motivated by the specular reflection in physics and illustrated in Figure 2. Figure 2a shows the clear correlation between the incident light and the reflected light [35]. Based on this concept, it is possible to compute a solution and its reverse solution in the manner depicted in Figure 2b. Based on the aforementioned occurrence, it may be concluded that a solution and its opposing solution corresponds to some extent. A superior solution could be discovered, if both solutions are compared. The superiority of the SRL method over OBL has been established [43,44]. Therefore, SRL is introduced during the initialization of the population of GSLEO-AEFA. The incorporation of GM allows for further mutation to be carried out on the population.



**Figure 2.** Specular reflection learning illustration; (a) The phenomenon of specular reflection and (b) The model of specular reflection.

Consider point  $P = (a, 0)$  in the horizontal plane and its opposite point  $P' = (b, 0)$ ,  $\forall P, P' \in [P_l, P_u]$ . The incidence angle is denoted by  $\alpha$  and the reflection angle is symbolized by  $\beta$ .  $O = (p_0, 0)$  represents the midpoint of  $[P_l, P_u]$ . From Figure 2b, the following equation can be deduced.

$$\tan(\alpha) = \frac{p_0 - a}{A_0} \tag{13}$$

$$\tan(\beta) = \frac{b - p_0}{B_0} \tag{14}$$

According to the reflection law  $\alpha$  and  $\beta$  are equal and the Equations (13) and (14) are represented as,

$$\frac{p_0 - a}{A_0} = \frac{b - p_0}{B_0} \tag{15}$$

From Equation (15),  $b$  is equal to either (16)–(18).

$$b = \frac{B_0(p_0 - a)}{A_0} + p_0 \tag{16}$$

If  $B_0 = \mu A_0$  then

$$b = \mu(p_0 - a) + p_0 \tag{17}$$

$$b = (\mu + 1)p_0 - \mu a = (0.5\mu + 0.5) * (P_l + P_u) - \mu a \tag{18}$$

where  $\mu$  denoted a preset factor for scaling and when it changes,  $b$  can be computed as given as follows.

$$b = \begin{cases} b_1, & \mu \in (0, 1), \\ 2p_0 - a, & \mu = 1 \\ b_2, & \mu \in (1, +\infty) \end{cases} \tag{19}$$

When  $\mu$  changes,  $b$  can explore all values of  $[P_l, P_u]$ . Hence, it may be used to initialize the population and improve its variety and exploration.

Let  $P = (p_1, p_2, \dots, p_n)$  represent an individual in a  $n$ -dimensional space. Individual  $p_i$  is within the range of  $[p_{\min}, p_{\max}]$  and  $i$  within the range of  $\{1, 2, \dots, n\}$  where  $n$  is the number of individuals. The opposing individual in Equation (12) can be deduced as follows with the in cooperation of the fundamental specular reflection law in Equation (18). Therefore, the opposing solution of  $p_i$  is expressed as follows,

$$p^* = (0.5\mu + 0.5) * (p_{\min} + p_{\max}) - \mu p_i \tag{20}$$

where  $\mu$  is a random number within the range of  $[0, 1]$ . Worthy of note is the fitness values of  $p_i$  and  $p^*$  must be compared to select the best population. It is generally known that meta-heuristic algorithms are significantly influenced by diversification of the population [45]. The use of SRL is justified because more diversity can make it possible for the search agents to expand the search area by resulting with the avoidance of suboptimal positions. To further enhance the diversification, GM is used to carry out mutation operations after creating the reverse solution.

$$p_{gi} = p^* * (1 + y * \text{Gaussian}(1)) \tag{21}$$

where Gaussian represents a Gaussian distribution with mean set 0 and variance set to 1, it returns a matrix of  $1 * 1$ ,  $y$  denotes a weight this is set to 1,  $p^*$  is an individual generated by SRL in Equation (20),  $p_{gi}$  denotes the newly generated point using Gaussian mutation.

### 2.2.2. Local Escaping Operator (LEO)

To improve the search capabilities of the gradient-based optimizer (GBO), local escaping operator (LEO) was initially developed by Iman et al. [46] by adjusting the locations of the individuals in relation to a certain criteria. LEO improves the quality of solutions. In particular, it enhances the convergence of the algorithm and keeps it from becoming stuck in local optima. It is worth to note here that local escaping operator (LEO) from the gradient-based optimizer (GBO) has been utilized to enhance the local escaping functionality of GSLEO-AEFA without altering the LEO itself. By incorporating the LEO within the algorithm, we leverage its ability to escape local optima and converge towards the global optimum, thereby improving the overall performance of the GSLEO-AEFA.

The best individual's position  $P_{\text{best}}$ , two randomly created individuals' positions  $P_{r1}^m$  and  $P_{r2}^m$ , two randomly selected positions  $P1_n^m$  and  $P2_n^m$ , and a new randomly generated solution  $P_k^m$  are used to generate a new solution  $P_{\text{LEO}}$ , which has great performance [15]. LEO formulas are expressed in Equations (22) and (24). If  $\text{rand} < 0.5$ , Equation (22) is computed, otherwise Equation (24).

$$P_{\text{LEO}} = P_n^m + f_1(u_1 P_{\text{best}} - u_2 P_k^m) + f_2 \rho_1 (u_3 (P2_n^m - P1_n^m)) + u_2 (P_{r1}^m - P_{r2}^m) / 2 \tag{22}$$

$$P_n^{m+1} = P_{\text{LEO}} \tag{23}$$

$$P_{LEO} = P_{best} + f_1(u_1 P_{best} - u_2 P_k^m) + f_2 \rho_1 (u_3 (P2_n^m - P1_n^m)) + u_2 (P_{r1}^m - P_{r2}^m) / 2 \quad (24)$$

$$P_n^{m+1} = P_{LEO} \quad (25)$$

where  $P_n^m$  is the position being modified by LEO,  $rand$  is a random number within the range of  $[0, 1]$ . Within the range of  $[-1, 1]$ ,  $f_1$  and  $f_2$  are random numbers uniformly distributed.

$$P1_n^m, P2_n^m = lb + rand(dim) \times (ub - lb) \quad (26)$$

$$u_1 = L_1 \times 2 \times rand + (1 - L_1) \quad (27)$$

$$u_2 = L_1 \times rand + (1 - L_1) \quad (28)$$

$$u_3 = L_1 \times rand + (1 - L_1) \quad (29)$$

where  $ub$  and  $lb$  denote the upper and lower boundaries, respectively,  $dim$  represents the solution dimension,  $n$  denotes the number of individuals in the population,  $m$  represents a decision variable in a given solution.  $u_1, u_2$ , and  $u_3$  are randomly generated variables. If  $k_1 < 0.5$ ,  $L_1$  set to 1, otherwise 0.  $k_1$  is number within the range  $[0, 1]$ .

$\rho_1$  in Equations (22) and (23) is expressed as follows,

$$\rho_1 = 2 \times rand \times \alpha - \alpha \quad (30)$$

$$\alpha = \left| \beta \times \sin\left(\frac{3\pi}{2} + \sin\left(\beta \times \frac{3\pi}{2}\right)\right) \right| \quad (31)$$

$$\beta = \beta_{min} + (\beta_{max} - \beta_{min}) \times \left(1 - \left(\frac{i}{i_{max}}\right)^3\right)^2 \quad (32)$$

where  $\beta_{min}$  and  $\beta_{max}$  are set to 0.2 and 1.2,  $i_{max}$  and  $i$  denote the maximum iteration and current iteration, respectively. To balance exploitation and exploration  $\rho_1$  is regulated by  $\alpha$ .

$P_k^m$  in Equations (22) and (23) is computed as follows:

$$P_k^m = \begin{cases} P_{rand} & \text{if } k_2 < 0.5 \\ P_r^m & \text{otherwise} \end{cases} \quad (33)$$

where  $P_{rand}$  is expressed in Equation (34),  $P_r^m$  is a random solution selected from the population.  $k_2$  denotes random float number between  $[0, 1]$ .

$$P_{rand} = lb + rand \times (ub - lb) \quad (34)$$

In Equation (35), if  $k_1 < 0.5$ ,  $L_2$  set to 1, otherwise 0.  $k_1$  is number within the range  $[0, 1]$ .

$$P_k^m = L_2 \times P_r^m + (1 - L_2) \times P_{rand} \quad (35)$$

Algorithm 2 and Figure 3 show the pseudocode and the flowchart of GSLEO-AEFA, respectively. The method begins with the population initialization using SRL and then uses GM to further increase the population diversity. The LEO is placed afterwards to improve the position of each particle.

**Algorithm 2** Gaussian Mutation Specular Reflection Learning with Local Escaping Operator Based Artificial Electric Field Algorithm (GSLEO-AEFA)

Initialize the population of  $p^*$  and  $p_{gi}$  using Equations (20) and (21)  
 Calculate the fitness of all particles initialized and select  $N$  best particles after ranking fitness from best to worst  
 Set the Initial  $Vel_i^t$  to 0  
 Calculate fitness value for all particles  
 Set iteration  $i \leftarrow 1$   
**while** stopping requirement is not met **do**  
     Compute  $K^t$ ,  $best(t)$  and  $worst(t)$   
     **for**  $i \leftarrow 1$  to  $N$  **do**  
         Calculate fitness values  
         Calculate the total force in each direction Equation (5)  
         Calculate acceleration Equation (3)  
          $Vel_i^{t+1} \leftarrow r \times Vel_i^t + accl_i^t$   
          $P_i^{t+1} \leftarrow P_i^t + Vel_i^t$   
     **if**  $rand < 0.5$  **then**  
         Update  $P_i^{t+1}$  using Equation (22)  
     **else**  
         Update  $P_i^{t+1}$  using Equation (24)  
     **end for**  
**end while**

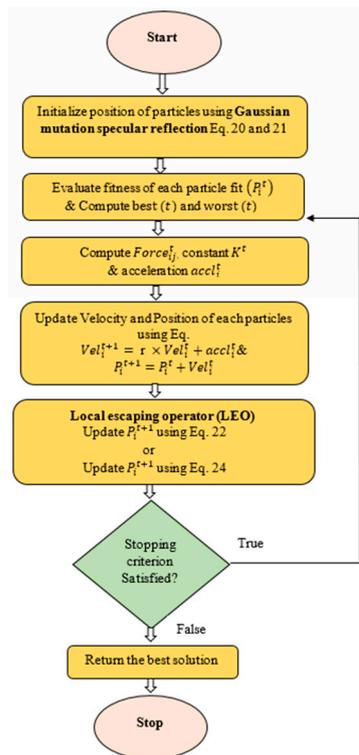


Figure 3. Flowchart of GSLEO-AEFA.

**3. Experimental Results and Discussion**

The suggested GSLEO-AEFA method is compared against six other algorithms, including AEFA [22], particle swarm optimization (PSO) algorithm [5], differential evolution (DE) algorithm [47], JAYA [48], sine-cosine algorithm (SCA) [13] and cuckoo search (CS) [49] to precisely assess its efficiency. 1000 iterations were completed with 30 executions to obtain the average and the standard deviation. The population size is set 30. Each algorithm’s parameters were chosen to reflect the original, first-published standard versions by the authors. The parameters for each algorithm are shown in Table 1.

**Table 1.** Parameter Settings.

Algorithms	Parameter Setting
GSLEO-AEFA	$k_0 = 500, \alpha = 30$
AEFA	$k_0 = 500, \alpha = 30$
PSO	$w_{\max} = 0.9, w_{\min} = 0.2, c_1 = c_2 = 2, v_{\max} = 6$
DE	$F = 0.5, CR = 0.7$
SCA	$a = 2$
CS	$Pa = 0.25, r = 0.05$

This test set consists of 10 fixed-dimension functions (F14–F23), six multi-modal functions (F8–F13), and seven single-modal functions (F1–F7). The detailed information is given in Table 2.

**Table 2.** Benchmark functions.

Function	Range	Dim	F <sub>min</sub>
$f_1(x) = \sum_{i=1}^n x_i^2$	[−100, 100]	30	0
$f_2(x) = \sum_{i=1}^n  x_i  + \prod_{i=1}^n  x_i $	[−10, 10]	30	0
$f_3(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j\right)^2$	[−100, 100]	30	0
$f_4(x) = \min\{ x_i , 1 \leq i \leq n\}$	[−100, 100]	30	0
$f_5(x) = \sum_{i=1}^{n-1} 100^i (x_{i+1} - x_i^2)^2 + (x_n - 1)^2$	[−30, 30]	30	0
$f_6(x) = \sum_{i=1}^n ((x_i + 0.5)^2)$	[−100, 100]	30	0
$f_7(x) = \sum_{i=1}^n ix_i^4 + \text{random}[0, 1]$	[−1.28, 1.28]	30	0
$f_8(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	[−500, 500]	30	$-418.9892 \times \text{dim}$
$f_9(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	[−5.12, 5.12]	30	0
$f_{10}(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	[−32, 32]	30	0
$f_{11}(x) = 1/4000 \sum_{i=1}^n \sum_{j=1}^n x_i^2 - \prod_{i=1}^n \cos(x_i / \sqrt{i}) + 1$	[−600, 600]	30	0
$f_{12}(x) = \pi/n \left\{ \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2 \right\} + \sum_{i=1}^n u(x_i, 10, 100, 4) + \pi/n 10 \sin(\pi y_1)$ $y_i = 1 + x_i + (1/4)u(x_i, a, k, m)$ $= \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	[−50, 50]	30	0
$f_{13}(x) = 0.1 \left\{ \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] \right\} + 0.1 \sin^2(3\pi x_1) + \sum_{i=1}^n u(x_i, 5, 100, 4)$	[−50, 50]	30	0
$f_{14}(x) = \left( (1/500) + \sum_{j=1}^{25} 1/j + \sum_{i=1}^2 (x_i - a_{ij})^6 \right)^{-1}$	[−65, 65]	2	1
$f_{15}(x) = \sum_{i=1}^{11} [a_i - x_1 (b_i^2 + b_i x_2) / b_i^2 + b_i x_3 + x_4]^2$	[−5, 5]	4	0.0003
$f_{16}(x) = 4x_1^2 - 2.1x_1^4 + 1/3x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	[−5, 5]	2	−1.0316
$f_{17}(x) = (x_2 - 5.1/4\pi^2 x_1^2 + 5/\pi x_1 - 6)^2 + 10(1 - (1/8\pi)) \cos x_1 + 10$	[−5, 5]	2	0.398
$f_{18}(x) = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	[−2, 2]	2	3
$f_{19}(x) = -\sum_{i=1}^4 c_i \exp\left[-\sum_{j=1}^3 a_{ij} (x_j - p_{ij})^2\right]$	[1, 3]	3	−3.86
$f_{20}(x) = -\sum_{i=1}^4 c_i \exp\left[-\sum_{j=1}^{j-1} a_{ij} (x_i - p_{ij})^2\right]$	[0, 1]	6	−3.32
$f_{21}(x) = -\sum_{i=1}^5 \left[ (X - a_i)(X - a_i)^T + c_i \right]^{-1}$	[0, 10]	4	−10.1532
$f_{22}(x) = -\sum_{i=1}^{F=1} \left[ (X - a_i)(X - a_i)^T + c_i \right]^{-1}$	[0, 10]	4	−10.4028
$f_{23}(x) = -\sum_{i=1}^{10} \left[ (X - a_i)(X - a_i)^T + c_i \right]^{-1}$	[0, 10]	4	−10.5363

### 3.1. Impact of Problem Dimension Analysis

To assess how effectively the algorithms worked in high dimensions, the F1–F13 functions in this study were expanded from 30 to 100 dimensions. The accuracy and stability of the algorithm were assessed using the mean and the standard deviation as the evaluation measures. Lower mean values and lower standard deviations indicate that the method is able to tackle the optimization problem well. Maximum iterations for all dimensions were set to 1000 and the total population size was set to 30.

The empirical results displayed in Table 3 for dimension set as 30. In order to determine the overall effectiveness (OE) of the algorithms, we utilized the formula developed by Qaraad et al. [50], which takes into account the number of Wins (W), Losses (L) and Ties (T) as outlined in Tables 3–5. The resulting OE values ranged from 84.62% to 92.31% with variation depending on the complexity of the problem in terms of dimension.

**Table 3.** F1-F23 Comparison with dimension = 30, significant values are in bold.

		GSLEO-AEFA	AEFA	PSO	DE	JAYA	SCA	CS
F1	AVG	<b><math>3.166 \times 10^{-76}</math></b>	$2.827 \times 10^{-21}$	$1.357 \times 10^{-8}$	$4.675 \times 10^{-13}$	$6.832 \times 10^{-12}$	$2.439 \times 10^{-2}$	$6.799 \times 10^{-3}$
	STD	<b><math>2.431 \times 10^{-77}</math></b>	$2.779 \times 10^{-21}$	$9.256 \times 10^{-10}$	$1.958 \times 10^{-14}$	$8.728 \times 10^{-15}$	$7.300 \times 10^{-3}$	$1.435 \times 10^{-3}$
F2	AVG	<b><math>1.071 \times 10^{-37}</math></b>	$1.156 \times 10^2$	5.334	$1.194 \times 10^{-7}$	$2.093 \times 10^{-9}$	$1.408 \times 10^{-5}$	$1.631 \times 10^{-1}$
	STD	<b><math>3.198 \times 10^{-38}</math></b>	$4.408 \times 10^1$	$6.310 \times 10^{-5}$	$5.981 \times 10^{-8}$	$2.115 \times 10^{-10}$	$3.339 \times 10^{-6}$	$5.914 \times 10^{-2}$
F3	AVG	<b><math>6.323 \times 10^{-72}</math></b>	$1.774 \times 10^3$	$1.414 \times 10^1$	$9.579 \times 10^3$	$1.211 \times 10^4$	$3.726 \times 10^3$	$3.379 \times 10^2$
	STD	<b><math>6.534 \times 10^{-73}</math></b>	$5.231 \times 10^2$	4.401	$4.586 \times 10^2$	$3.148 \times 10^3$	$1.379 \times 10^3$	$5.314 \times 10^1$
F4	AVG	<b><math>5.486 \times 10^{-38}</math></b>	1.314	$6.036 \times 10^{-1}$	$1.068 \times 10^1$	$1.041 \times 10^1$	$1.925 \times 10^1$	5.726
	STD	<b><math>1.030 \times 10^{-38}</math></b>	1.196	$8.508 \times 10^{-2}$	5.046	6.554	$1.208 \times 10^1$	3.251
F5	AVG	<b><math>2.635 \times 10^1</math></b>	$4.102 \times 10^2$	$5.220 \times 10^1$	$3.564 \times 10^1$	$9.954 \times 10^2$	$4.391 \times 10^2$	$4.301 \times 10^1$
	STD	$2.495 \times 10^{-1}$	$4.270 \times 10^2$	7.801	<b><math>1.369 \times 10^{-1}</math></b>	$1.519 \times 10^{-1}$	$5.683 \times 10^{-1}$	$3.175 \times 10^1$
F6	AVG	<b><math>1.751 \times 10^{-21}</math></b>	$2.148 \times 10^{-21}$	$1.258 \times 10^{-8}$	$2.671 \times 10^{-13}$	3.284	4.519	$7.427 \times 10^{-3}$
	STD	<b><math>2.239 \times 10^{-22}</math></b>	$5.626 \times 10^{-22}$	$1.805 \times 10^{-9}$	$1.590 \times 10^{-14}$	$1.952 \times 10^{-1}$	$9.424 \times 10^{-2}$	$9.015 \times 10^{-5}$
F7	AVG	<b><math>3.495 \times 10^{-4}</math></b>	2.632	6.237	$1.958 \times 10^{-2}$	$2.528 \times 10^{-2}$	$3.961 \times 10^{-2}$	$4.488 \times 10^{-2}$
	STD	<b><math>4.928 \times 10^{-5}</math></b>	$1.781 \times 10^{-1}$	$4.678 \times 10^{-3}$	$1.513 \times 10^{-3}$	$8.613 \times 10^{-3}$	$3.066 \times 10^{-2}$	$1.219 \times 10^{-2}$
F8	AVG	$-2.483 \times 10^5$	<b><math>-7.901 \times 10^4</math></b>	$-1.922 \times 10^5$	$-2.023 \times 10^5$	$-1.686 \times 10^5$	$-1.158 \times 10^5$	$-3.243 \times 10^5$
	STD	$-8.278 \times 10^3$	<b><math>-2.634 \times 10^3</math></b>	$-6.406 \times 10^3$	$-6.742 \times 10^3$	$-5.619 \times 10^3$	$-3.859 \times 10^3$	$-1.127 \times 10^4$
F9	AVG	<b>0</b>	$4.554 \times 10^1$	$9.836 \times 10^1$	$1.639 \times 10^2$	$6.256 \times 10^1$	$1.884 \times 10^1$	$7.841 \times 10^1$
	STD	<b>0</b>	$9.949 \times 10^1$	$2.097 \times 10^1$	$1.246 \times 10^1$	$2.709 \times 10^1$	$1.204 \times 10^{-1}$	$1.165 \times 10^1$
F10	AVG	<b><math>4.441 \times 10^{-16}</math></b>	$3.164 \times 10^{-1}$	$5.499 \times 10^{-2}$	$2.089 \times 10^{-7}$	$1.407 \times 10^{-7}$	$1.165 \times 10^1$	1.152
	STD	<b>0</b>	0	$8.231 \times 10^{-1}$	$6.441 \times 10^{-9}$	$2.748 \times 10^{-8}$	$1.008 \times 10^1$	$4.479 \times 10^{-1}$
F11	AVG	<b>0</b>	$5.602 \times 10^{-1}$	$9.275 \times 10^{-3}$	$2.053 \times 10^{-3}$	$1.157 \times 10^{-2}$	$2.772 \times 10^{-1}$	$8.599 \times 10^{-2}$
	STD	<b>0</b>	$7.477 \times 10^{-2}$	$7.390 \times 10^{-3}$	$2.460 \times 10^{-4}$	$7.472 \times 10^{-14}$	$1.884 \times 10^{-1}$	$7.931 \times 10^{-2}$
F12	AVG	<b><math>1.291 \times 10^{-23}</math></b>	2.225	$3.482 \times 10^{-11}$	$2.729 \times 10^{-12}$	$6.147 \times 10^{-1}$	$1.519 \times 10^1$	1.652
	STD	<b><math>6.678 \times 10^{-24}</math></b>	$2.885 \times 10^{-1}$	$2.243 \times 10^{-11}$	$1.113 \times 10^{-13}$	$1.271 \times 10^{-2}$	1.020	$7.550 \times 10^{-1}$
F13	AVG	$3.296 \times 10^{-3}$	6.300	$5.127 \times 10^{-3}$	<b><math>2.695 \times 10^{-3}</math></b>	$2.957 \times 10^4$	$2.185 \times 10^2$	$7.119 \times 10^{-1}$
	STD	<b>0</b>	$7.164 \times 10^{-1}$	$1.099 \times 10^{-3}$	$1.084 \times 10^{-3}$	1.166	$3.592 \times 10^{-1}$	$3.981 \times 10^{-1}$
F14	AVG	<b><math>9.980 \times 10^{-1}</math></b>	2.985	4.763	<b><math>9.980 \times 10^{-1}</math></b>	1.065	1.263	<b><math>9.980 \times 10^{-1}</math></b>
	STD	<b>0</b>	1.149	2.456	<b>0</b>	$1.062 \times 10^{-3}$	$7.200 \times 10^{-4}$	<b>0</b>
F15	AVG	<b><math>3.109 \times 10^{-4}</math></b>	$3.054 \times 10^{-3}$	$4.344 \times 10^{-3}$	$1.826 \times 10^{-3}$	$1.119 \times 10^{-3}$	$9.437 \times 10^{-4}$	$3.289 \times 10^{-4}$
	STD	$1.019 \times 10^{-5}$	$6.377 \times 10^{-5}$	$1.540 \times 10^{-4}$	$2.196 \times 10^{-4}$	$2.765 \times 10^{-4}$	$3.308 \times 10^{-5}$	<b><math>6.779 \times 10^{-6}</math></b>
F16	AVG	<b>-1.032</b>	<b>-1.032</b>	<b>-1.032</b>	<b>-1.032</b>	<b>-1.032</b>	<b>-1.032</b>	<b>-1.032</b>
	STD	$4.441 \times 10^{-16}$	<b>0</b>	<b>0</b>	<b>0</b>	$5.000 \times 10^{-7}$	$3.000 \times 10^{-6}$	<b>0</b>
F17	AVG	<b><math>3.979 \times 10^{-1}</math></b>	<b><math>3.979 \times 10^{-1}</math></b>	<b><math>3.979 \times 10^{-1}</math></b>	<b><math>3.979 \times 10^{-1}</math></b>	<b><math>3.979 \times 10^{-1}</math></b>	$3.985 \times 10^{-1}$	<b><math>3.979 \times 10^{-1}</math></b>
	STD	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	$8.784 \times 10^{-4}$	<b>0</b>
F18	AVG	<b>3</b>	<b>3</b>	<b>3</b>	<b>3</b>	3.284	<b>3</b>	<b>3</b>
	STD	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	$4.250 \times 10^{-4}$	$5.500 \times 10^{-6}$	<b>0</b>
F19	AVG	-3.863	-3.777	<b>-3.861</b>	-3.863	-3.667	-3.855	-3.863
	STD	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	$1.088 \times 10^{-1}$	$2.215 \times 10^{-3}$	<b>0</b>
F20	AVG	-3.314	-1.646	-3.237	-3.259	-1.957	-2.886	<b>-3.322</b>
	STD	<b>0</b>	$3.028 \times 10^{-1}$	$1.527 \times 10^{-1}$	$5.945 \times 10^{-2}$	$1.353 \times 10^{-1}$	$1.960 \times 10^{-1}$	<b>0</b>
F21	AVG	<b><math>-1.015 \times 10^1</math></b>	-6.101	-8.294	-8.652	-1.379	-2.450	<b><math>-1.015 \times 10^1</math></b>
	STD	<b>0</b>	3.735	3.735	0	1.265	$5.761 \times 10^{-1}$	<b>0</b>
F22	AVG	<b><math>-1.040 \times 10^1</math></b>	$-1.010 \times 10^1$	-9.621	$-1.023 \times 10^1$	-1.854	-3.455	<b><math>-1.040 \times 10^1</math></b>
	STD	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	1.559	1.078	<b>0</b>

Table 3. Cont.

		GSLEO-AEFA	AEFA	PSO	DE	JAYA	SCA	CS
F23	AVG	$-1.054 \times 10^1$	$-1.054 \times 10^1$	-9.862	$-1.031 \times 10^1$	-2.339	-4.853	$-1.054 \times 10^1$
	STD	<b>0</b>	<b>0</b>	4.057	<b>0</b>	$7.908 \times 10^{-2}$	$4.472 \times 10^{-1}$	<b>0</b>
	W/L/T OE	12/3/8 86.95%	1/18/4 21.73%	0/19/4 17.39%	1/17/5 26.08%	0/22/1 4.34%	0/23/0 0%	1/14/8 39.13%

Table 4. F1–F13 Comparison with dimension = 50, significant values are in bold.

		GSLEO-AEFA	AEFA	PSO	DE	JAYA	SCA	CS
F1	AVG	$1.312 \times 10^{-75}$	$1.311 \times 10^1$	$3.983 \times 10^{-3}$	$6.081 \times 10^{-6}$	$9.955 \times 10^{-8}$	$1.376 \times 10^2$	2.774
	STD	$6.044 \times 10^{-76}$	$3.475 \times 10^{-1}$	$1.564 \times 10^{-4}$	$4.101 \times 10^{-6}$	$5.847 \times 10^{-9}$	$1.154 \times 10^1$	1.152
F2	AVG	$1.620 \times 10^{-37}$	$2.192 \times 10^2$	$2.681 \times 10^1$	$1.064 \times 10^{-3}$	$1.191 \times 10^{-6}$	$1.124 \times 10^{-2}$	1.990
	STD	$2.567 \times 10^{-37}$	$6.187 \times 10^1$	5.091	$7.480 \times 10^{-4}$	$8.203 \times 10^{-7}$	$2.452 \times 10^{-3}$	$5.229 \times 10^{-1}$
F3	AVG	$1.773 \times 10^{-69}$	$5.275 \times 10^3$	$6.116 \times 10^2$	$7.020 \times 10^4$	$6.268 \times 10^4$	$3.841 \times 10^4$	$3.916 \times 10^3$
	STD	$9.445 \times 10^{-69}$	$8.081 \times 10^2$	$1.721 \times 10^2$	$9.668 \times 10^3$	$7.966 \times 10^2$	$7.215 \times 10^3$	$1.412 \times 10^1$
F4	AVG	$5.798 \times 10^{-38}$	8.926	2.353	$5.746 \times 10^1$	$5.065 \times 10^1$	$5.776 \times 10^1$	$1.445 \times 10^1$
	STD	$7.141 \times 10^{-38}$	$8.845 \times 10^{-1}$	$2.473 \times 10^{-1}$	$2.310 \times 10^1$	1.968	4.991	$7.853 \times 10^{-1}$
F5	AVG	$4.698 \times 10^1$	$5.315 \times 10^4$	$1.652 \times 10^2$	$1.685 \times 10^3$	$9.021 \times 10^3$	$1.438 \times 10^6$	$6.367 \times 10^2$
	STD	$4.170 \times 10^{-1}$	$3.820 \times 10^4$	$3.482 \times 10^1$	$2.009 \times 10^1$	$1.645 \times 10^{-1}$	$2.404 \times 10^5$	$2.091 \times 10^2$
F6	AVG	$7.007 \times 10^{-3}$	$1.113 \times 10^1$	$2.008 \times 10^{-3}$	$6.021 \times 10^{-6}$	7.942	$1.409 \times 10^2$	2.841
	STD	$1.019 \times 10^{-4}$	$5.086 \times 10^{-1}$	$7.776 \times 10^{-5}$	$2.923 \times 10^{-6}$	$8.489 \times 10^{-1}$	$3.747 \times 10^1$	$7.485 \times 10^{-1}$
F7	AVG	$2.918 \times 10^{-4}$	$4.201 \times 10^2$	$4.192 \times 10^1$	$6.239 \times 10^{-2}$	$6.597 \times 10^{-2}$	$5.376 \times 10^{-1}$	$1.918 \times 10^{-1}$
	STD	$1.265 \times 10^{-4}$	$2.671 \times 10^1$	$1.212 \times 10^1$	$1.567 \times 10^{-2}$	$1.756 \times 10^{-2}$	$3.018 \times 10^{-1}$	$6.508 \times 10^{-2}$
F8	AVG	$-1.177 \times 10^4$	$-3.109 \times 10^3$	$-1.020 \times 10^4$	$-8.514 \times 10^3$	$-6.917 \times 10^3$	$-5.053 \times 10^3$	$-1.381 \times 10^4$
	STD	$1.283 \times 10^2$	$2.806 \times 10^2$	$3.958 \times 10^2$	$6.519 \times 10^2$	$1.020 \times 10^2$	$3.026 \times 10^2$	$1.781 \times 10^3$
F9	AVG	<b>0</b>	$1.906 \times 10^2$	$2.523 \times 10^2$	$3.497 \times 10^2$	$1.150 \times 10^2$	$5.247 \times 10^1$	$1.637 \times 10^2$
	STD	<b>0</b>	$2.702 \times 10^1$	1.893	$1.600 \times 10^1$	$3.186 \times 10^1$	3.542	$1.172 \times 10^1$
F10	AVG	$4.441 \times 10^{-16}$	3.316	$6.905 \times 10^{-1}$	$6.402 \times 10^{-1}$	$4.707 \times 10^{-5}$	$1.824 \times 10^1$	3.490
	STD	<b>0</b>	$2.252 \times 10^{-1}$	$2.308 \times 10_{-1}$	$1.130 \times 10^{-3}$	$2.021 \times 10^{-5}$	9.498	$9.095 \times 10^{-2}$
F11	AVG	<b>0</b>	7.215	$5.453 \times 10^{-3}$	$2.384 \times 10^{-3}$	$6.795 \times 10^{-3}$	2.092	$9.397 \times 10^{-1}$
	STD	<b>0</b>	1.062	$3.695 \times 10^{-3}$	$9.500 \times 10^{-7}$	$9.182 \times 10^{-4}$	$8.224 \times 10^{-1}$	$1.529 \times 10^{-1}$
F12	AVG	$2.584 \times 10^{-3}$	7.278	$2.718 \times 10^{-2}$	$6.784 \times 10^{-2}$	1.357	$2.117 \times 10^6$	3.802
	STD	$1.779 \times 10^{-4}$	1.109	$2.290 \times 10^{-5}$	$5.922 \times 10^{-2}$	$6.044 \times 10^{-2}$	$4.215 \times 10^5$	$7.237 \times 10^{-1}$
F13	AVG	$6.550 \times 10^{-2}$	$1.240 \times 10^2$	$7.207 \times 10^{-3}$	2.900	$3.611 \times 10^3$	$1.074 \times 10^7$	$3.649 \times 10^1$
	STD	$1.586 \times 10^{-3}$	$1.541 \times 10^2$	$8.781 \times 10^{-4}$	$2.738 \times 10^{-4}$	4.381	$6.499 \times 10^4$	$1.280 \times 10^1$
	W/L/T	11/2/0	0/13/0	1/12/0	1/12/0	0/13/0	0/13/0	0/13/0
	OE	<b>84.62%</b>	0%	7.69%	7.69%	0%	0%	0%

The data make it clear that GSLEO-AEFA receives the near optimal solution in seven of the seven single-modal functions (F1–F7). It is absolutely critical to note that the achievement of GSLEO-AEFA is a significant improvement over the original AEFA and other algorithms in F1–F7. This improvement is expected due to the introduction of LEO, which improves GSLEO-AEFAs ability to do local search. As a result, there is significant improvement in exploitation too.

GSLEO-AEFA achieves the best performance on F9–F12 and F14–F23 in the multimodal benchmark functions (F8–F23). Each of the results obtained by GSLEO-AEFA is better than what the initial AEFA is able to achieve. Global search capacity and population diversity of GSLEO-AEFA in comparison to the initial AEFA are drastically improved. Additionally, the demonstration of multimodal functions with fixed dimension characteristics proves that the new mechanism introduced successfully achieves a good balance between global search and local exploitation.

Tables 4 and 5 display the results of expanding the dimension from 50 and 100, respectively. The tables present the evidence that as the problem’s dimension increases, it becomes less easy to find optimum solutions. Table 4 clearly shows that GSLEO-AEFA achieves the best performance in F1–F5, F7 and F9–F12 despite the growth in dimension. GSLEO-AEFA achieves the best results in thirteen benchmark function when dimension is 100 in Table 5. When the results from the tables are analyzed, it can be seen that GSLEO-AEFA is quite robust for handling complex problems. This indicates that population

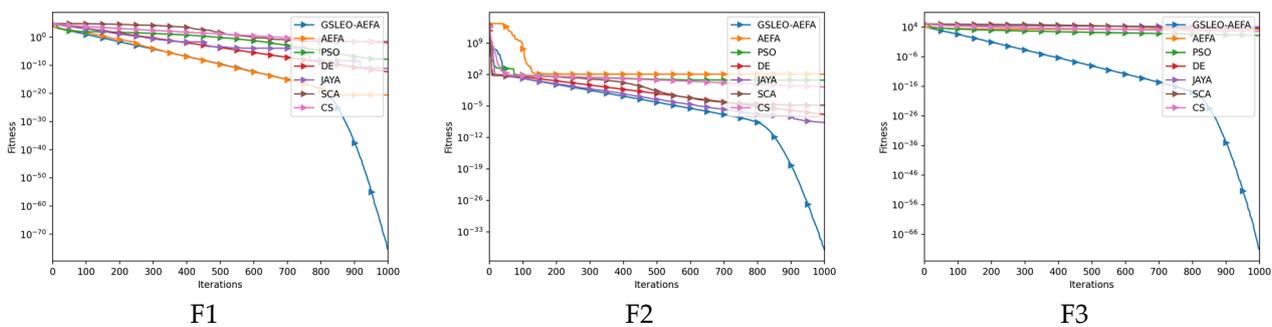
diversity improved based on proposed GM based SRL. Additionally, LEO is used to update the positions of particles, so that they continually progress in the direction of the global best, achieved early exploration and late exploitation.

**Table 5.** F1–F13 Comparison with dimension = 100, significant values are in bold.

		GSLEO-AEFA	AEFA	PSO	DE	JAYA	SCA	CS
F1	AVG	<b>1.825 × 10<sup>-73</sup></b>	1.091 × 10 <sup>3</sup>	3.321	4.915	1.084 × 10 <sup>2</sup>	5.953 × 10 <sup>3</sup>	3.227 × 10 <sup>2</sup>
	STD	<b>3.217 × 10<sup>-74</sup></b>	3.016 × 10 <sup>2</sup>	2.080	3.551	5.745 × 10 <sup>-3</sup>	6.759 × 10 <sup>2</sup>	5.338 × 10 <sup>1</sup>
F2	AVG	<b>2.717 × 10<sup>-37</sup></b>	4.315 × 10 <sup>2</sup>	1.135 × 10 <sup>2</sup>	1.405	1.336 × 10 <sup>-3</sup>	1.799	1.579 × 10 <sup>1</sup>
	STD	<b>2.624 × 10<sup>-37</sup></b>	1.948 × 10 <sup>1</sup>	8.413	6.684 × 10 <sup>-1</sup>	3.283 × 10 <sup>-4</sup>	6.289 × 10 <sup>-3</sup>	5.369 × 10 <sup>-1</sup>
F3	AVG	<b>6.493 × 10<sup>-71</sup></b>	1.600 × 10 <sup>4</sup>	1.097 × 10 <sup>4</sup>	3.346 × 10 <sup>5</sup>	2.863 × 10 <sup>5</sup>	1.876 × 10 <sup>5</sup>	3.266 × 10 <sup>4</sup>
	STD	<b>2.866 × 10<sup>-72</sup></b>	7.989 × 10 <sup>2</sup>	8.134 × 10 <sup>2</sup>	3.184 × 10 <sup>4</sup>	7.466 × 10 <sup>4</sup>	1.587 × 10 <sup>3</sup>	2.355 × 10 <sup>3</sup>
F4	AVG	<b>3.877 × 10<sup>-37</sup></b>	1.662 × 10 <sup>1</sup>	9.432	9.636 × 10 <sup>1</sup>	9.338 × 10 <sup>1</sup>	8.730 × 10 <sup>1</sup>	2.282 × 10 <sup>1</sup>
	STD	<b>6.640 × 10<sup>-38</sup></b>	1.495	2.698 × 10 <sup>-1</sup>	6.600 × 10 <sup>-1</sup>	6.479 × 10 <sup>-1</sup>	8.400 × 10 <sup>-1</sup>	2.082
F5	AVG	<b>9.737 × 10<sup>1</sup></b>	2.570 × 10 <sup>6</sup>	5.257 × 10 <sup>3</sup>	1.481 × 10 <sup>4</sup>	7.654 × 10 <sup>3</sup>	6.989 × 10 <sup>7</sup>	7.093 × 10 <sup>4</sup>
	STD	<b>4.531 × 10<sup>-1</sup></b>	2.367 × 10 <sup>3</sup>	7.594 × 10 <sup>1</sup>	5.708 × 10 <sup>2</sup>	3.352 × 10 <sup>3</sup>	1.956 × 10 <sup>7</sup>	2.230 × 10 <sup>4</sup>
F6	AVG	<b>2.314</b>	1.067 × 10 <sup>3</sup>	2.575	3.749	2.248 × 10 <sup>1</sup>	5.981 × 10 <sup>3</sup>	3.865 × 10 <sup>2</sup>
	STD	<b>1.672 × 10<sup>-1</sup></b>	4.358 × 10 <sup>1</sup>	1.103	1.543	6.895 × 10 <sup>-1</sup>	1.614 × 10 <sup>3</sup>	5.581 × 10 <sup>1</sup>
F7	AVG	<b>2.347 × 10<sup>-4</sup></b>	1.842 × 10 <sup>3</sup>	3.258 × 10 <sup>2</sup>	5.446 × 10 <sup>-1</sup>	4.352 × 10 <sup>-1</sup>	4.594 × 10 <sup>1</sup>	1.285
	STD	<b>6.276 × 10<sup>-5</sup></b>	7.705 × 10 <sup>1</sup>	1.977	6.282 × 10 <sup>-2</sup>	1.667 × 10 <sup>-1</sup>	9.523	4.334 × 10 <sup>-1</sup>
F8	AVG	<b>-2.052 × 10<sup>4</sup></b>	-5.031 × 10 <sup>3</sup>	-1.824 × 10 <sup>4</sup>	<b>-1.176 × 10<sup>4</sup></b>	-1.053 × 10 <sup>4</sup>	-7.068 × 10 <sup>3</sup>	-1.954 × 10 <sup>4</sup>
	STD	<b>2.074 × 10<sup>2</sup></b>	1.335 × 10 <sup>3</sup>	3.748 × 10 <sup>2</sup>	6.889 × 10 <sup>2</sup>	1.966 × 10 <sup>3</sup>	4.924 × 10 <sup>2</sup>	1.082 × 10 <sup>3</sup>
F9	AVG	<b>0</b>	8.151 × 10 <sup>2</sup>	7.145 × 10 <sup>2</sup>	8.851 × 10 <sup>2</sup>	3.946 × 10 <sup>2</sup>	2.222 × 10 <sup>2</sup>	4.332 × 10 <sup>2</sup>
	STD	<b>0</b>	9.659 × 10 <sup>1</sup>	5.014 × 10 <sup>1</sup>	1.278 × 10 <sup>1</sup>	1.397 × 10 <sup>1</sup>	2.628 × 10 <sup>1</sup>	1.417
F10	AVG	<b>4.441 × 10<sup>-16</sup></b>	8.082	2.605	4.815	1.090 × 10 <sup>-2</sup>	1.971 × 10 <sup>1</sup>	6.946
	STD	<b>0</b>	8.989 × 10 <sup>-2</sup>	2.132 × 10 <sup>-1</sup>	2.543	2.179 × 10 <sup>-3</sup>	5.400 × 10 <sup>-3</sup>	6.168 × 10 <sup>-1</sup>
F11	AVG	<b>0</b>	4.747 × 10 <sup>1</sup>	4.632 × 10 <sup>-2</sup>	7.983 × 10 <sup>-1</sup>	6.488 × 10 <sup>-2</sup>	4.328 × 10 <sup>1</sup>	4.429
	STD	<b>0</b>	2.710	3.568 × 10 <sup>-3</sup>	2.028 × 10 <sup>-1</sup>	9.182 × 10 <sup>-4</sup>	7.405	7.023 × 10 <sup>-1</sup>
F12	AVG	<b>3.018 × 10<sup>-2</sup></b>	6.694 × 10 <sup>1</sup>	2.211	1.185 × 10 <sup>4</sup>	5.679 × 10 <sup>4</sup>	1.592 × 10 <sup>8</sup>	1.092 × 10 <sup>1</sup>
	STD	<b>1.485 × 10<sup>-2</sup></b>	4.174 × 10 <sup>1</sup>	1.714	2.051 × 10 <sup>3</sup>	1.489	4.429 × 10 <sup>7</sup>	1.461
F13	AVG	<b>2.857</b>	2.595 × 10 <sup>5</sup>	1.454 × 10 <sup>1</sup>	8.643 × 10 <sup>4</sup>	5.953 × 10 <sup>4</sup>	2.934 × 10 <sup>8</sup>	5.143 × 10 <sup>3</sup>
	STD	<b>3.944 × 10<sup>-1</sup></b>	1.496 × 10 <sup>5</sup>	6.216	3.281 × 10 <sup>4</sup>	3.263 × 10 <sup>2</sup>	2.322 × 10 <sup>8</sup>	2.547 × 10 <sup>3</sup>
	<b>W/L/T</b>	12/1/0	0/13/0	0/13/0	1/12/0	0/13/0	0/13/0	0/13/0
	<b>OE</b>	92.31%	0%	0%	7.69%	0%	0%	0%

### 3.2. Convergence Trajectory Analysis

In Figure 4, with F1–F13 set to 30 dimensions, displays the convergence curves of seven different methods. GSLEO-AEFA converged to the global optimum on F9–F11, F16–F17, and F16–F23 and exhibited a greater convergence rate than compared algorithms in multi-modal functions. GSLEO-AEFA significantly increased the convergence accuracy for unimodal functions as compared to the original AEFA (F1–F7). The effective implementation of the LEO mechanism and application of changes to the initial population generation approach ensure that GSLEO-AEFA has enhanced optimization and convergence performance.



**Figure 4.** Cont.

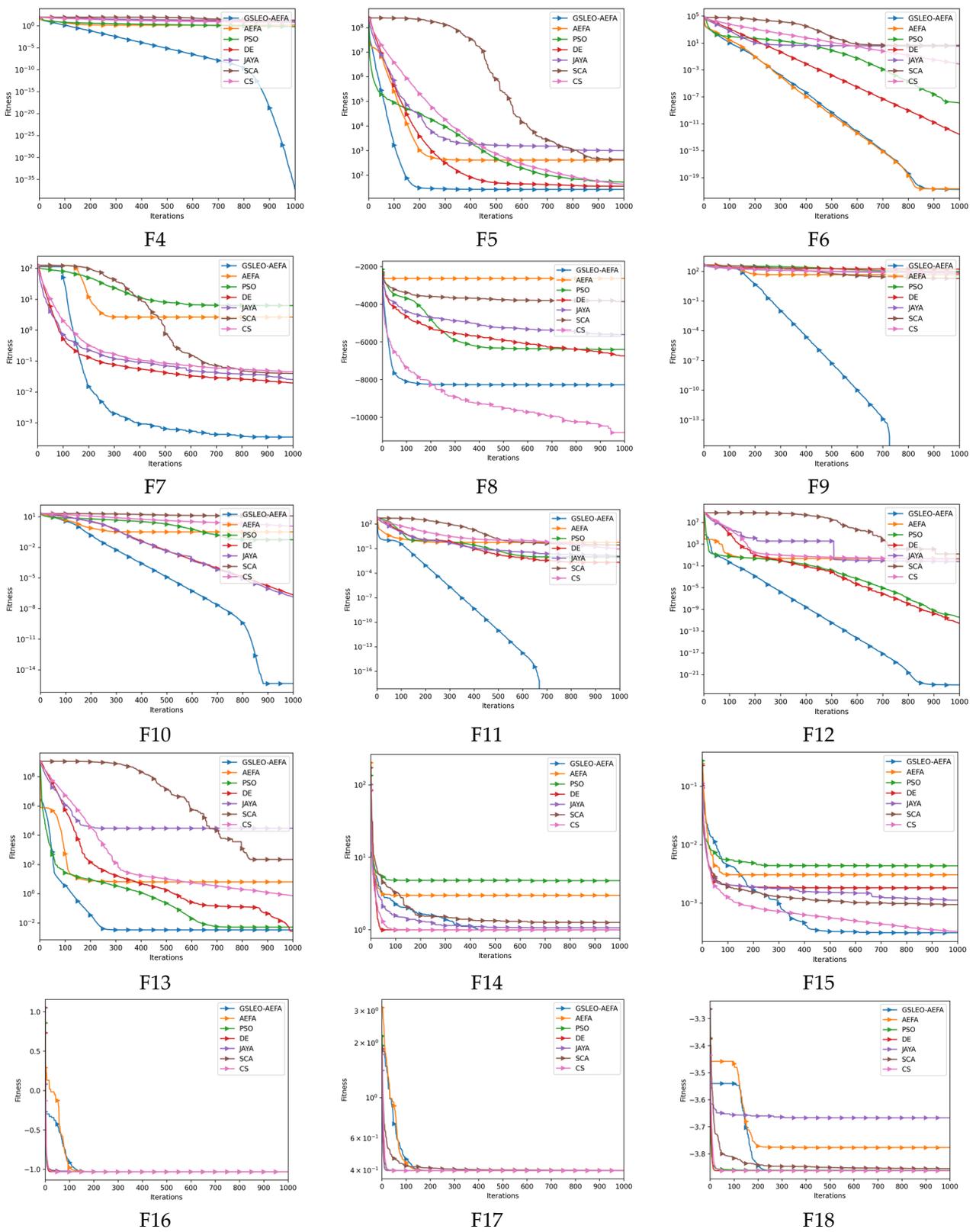
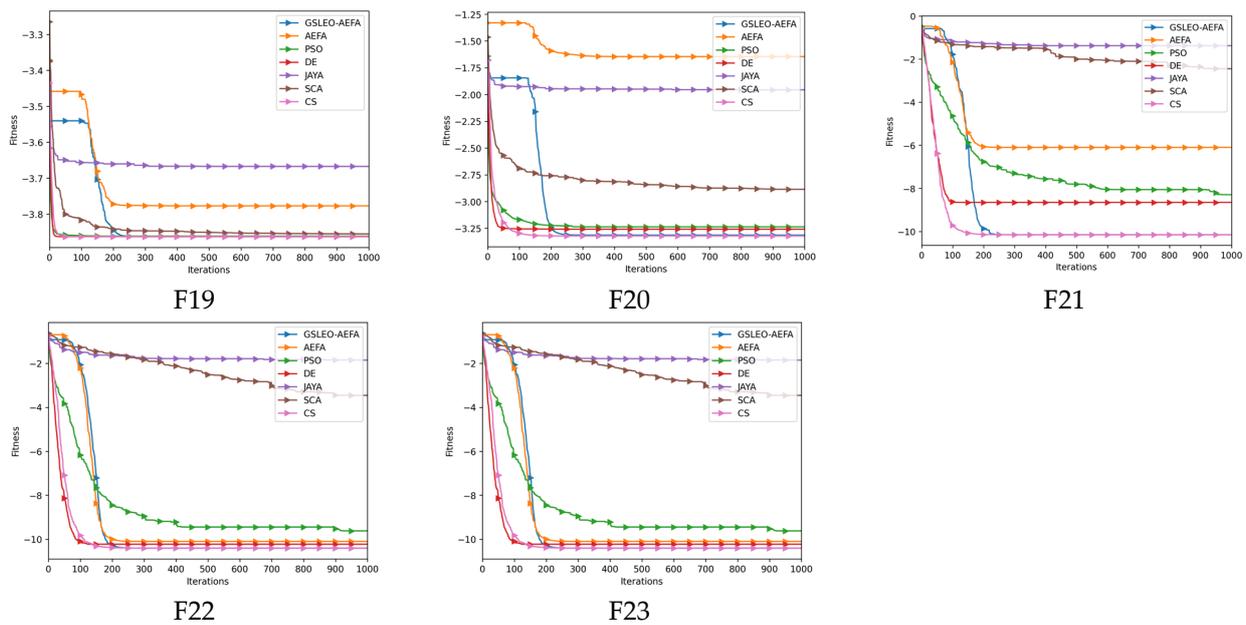


Figure 4. Cont.



**Figure 4.** Convergence trajectory graphs for all studied benchmark functions F1–F23.

### 3.3. Statistical Test

It is suggested that not only mean and standard deviation comparisons but more evaluation metrics should be used to assess the effectiveness of the metaheuristic algorithms [51]. The use of statistical tests such as the Friedman test and Wilcoxon rank-sum test are advised as effective evaluation metrics to compare the performance of different algorithms.

The significance threshold for the Wilcoxon rank-sum test was set at 0.05. The approach was deemed statistically better, when  $p < 0.05$ . Table 6 displayed the Wilcoxon rank-sum test findings with the notation “+/-/=” denoting whether the recommended approach was superior to, inferior to, or equal to the compared approach [52]. According to Table 6, GSLEO-AEFA always provides R+ values that are greater than R– values. Additionally, Table 6 reveals that the  $p$  values of the six algorithms are less than 0.05, indicating that GSLEO-AEFA is superior to the others. The + value of GSLEO-AEFA increased as the dimension increased from 50 to 100, as shown in Table 6. This indicates that the GSLEO-AEFA algorithm performs better than other algorithms, as the dimension increases. The results show that GSLEO-AEFA has higher level of solution accuracy.

**Table 6.** Wilcoxon test analysis results.

Dimension	Case	–	+	=	R–	R+	$p$ -Value
30	GSLEO-AEFA vs. AEFA	0	19	4	0	190	$1.32 \times 10^{-4}$
	GSLEO-AEFA vs. PSO	0	19	4	0	190	$1.32 \times 10^{-4}$
	GSLEO-AEFA vs. DE	1	17	5	6	165	$5.35 \times 10^{-4}$
	GSLEO-AEFA vs. JAYA	0	21	2	0	231	$6.00 \times 10^{-5}$
	GSLEO-AEFA vs. SCA	0	20	3	0	210	$8.90 \times 10^{-5}$
	GSLEO-AEFA vs. CS	2	13	8	19	101	$1.99 \times 10^{-2}$
50	GSLEO-AEFA vs. AEFA	0	13	0	0	91	$1.32 \times 10^{-4}$
	GSLEO-AEFA vs. PSO	2	11	0	7	84	$1.32 \times 10^{-4}$
	GSLEO-AEFA vs. DE	1	12	0	7	87	$5.35 \times 10^{-4}$
	GSLEO-AEFA vs. JAYA	0	13	0	0	91	$6.00 \times 10^{-5}$
	GSLEO-AEFA vs. SCA	0	13	0	0	91	$8.90 \times 10^{-5}$
	GSLEO-AEFA vs. CS	1	12	0	12	79	$1.99 \times 10^{-2}$
100	GSLEO-AEFA vs. AEFA	0	13	0	0	91	$1.47 \times 10^{-3}$
	GSLEO-AEFA vs. PSO	0	13	0	0	91	$1.47 \times 10^{-3}$
	GSLEO-AEFA vs. DE	0	13	0	0	91	$1.47 \times 10^{-3}$
	GSLEO-AEFA vs. JAYA	0	13	0	0	91	$1.47 \times 10^{-3}$
	GSLEO-AEFA vs. SCA	0	13	0	0	91	$1.47 \times 10^{-3}$
	GSLEO-AEFA vs. CS	0	13	0	0	91	$1.47 \times 10^{-3}$

The validity of the experiments is evaluated using the Friedman test by contrasting the proposed GSLEO-AEFA with other algorithms. The Friedman test is one of the widely used statistical tests which is used to determine whether more than one techniques' outputs differ significantly [53]. The outcomes of the Friedman tests are presented in Table 7. The technique that obtains the lowest rank is the most efficient according to Friedman test. In each of the various scenarios, the suggested GSLEO-AEFA is always rated first. Table 7 shows the results for the 30, 50 and 100 dimensions. The GSLEO-AEFA outperforms other algorithms in terms of competitiveness.

**Table 7.** Friedman test for dimension 30, 50, and 100 dimensions for 20 functions.

Test	Dimension	GSLEO-AEFA	AEFA	PSO	DE	JAYA	SCA	CS
Friedman value	30	1.7	4.74	4.3	3.35	5.07	5.33	3.52
Friedman rank		1	5	4	2	6	7	3
Friedman value	50	1.31	5.69	3.31	3.69	4	5.92	4.08
Friedman rank		1	6	2	3	4	7	5
Friedman value	100	1	5.69	3.08	4.62	3.69	5.77	4.15
Friedman rank		1	6	2	5	3	7	4

In conclusion, AEFA-CSR is more competitive than innovative algorithms such as SCA and CS as well as traditional algorithms such as DE and PSO. The greater accomplishments of GSLEO-AEFA can be attributed to the newly implemented strategies. The LEO solution strategy makes the local optimal escape mechanism better and the GM based SRL makes the population more diversified. As a result, when the two methods are combined, the capacity of GSLEO-AEFA to solve multimodal and unimodal functions is significantly improved.

### 3.4. Diversity Analysis

In order to comprehensively evaluate the performance of the GSLEO-AEFA optimization algorithm, it is imperative to thoroughly investigate the impact of Gaussian mutation and specular reflection on its diversity. To accomplish this, we must calculate the population diversity of both GSLEO-AEFA and AEFA using the following equation [20].

$$Diversity = \frac{1}{N} \sum_{i=1}^N \sqrt{\sum_{j=1}^D (x_{i,j} - \bar{x}_j)^2}, \bar{x}_j = \frac{\sum_{i=1}^N x_{i,j}}{N} \quad (36)$$

Here,  $D$  denotes dimension,  $N$  represents the number of particles,  $x_j$  is  $j$ th dimension for the particle at the center of the population and  $x_{i,j}$  represents  $j$ th dimension of a given particle  $i$ . To validate the success of GSLEO-AEFA on diversity over the algorithm which is originated from, diversity plots for two unimodal; F1 and F2 and two multimodal; F10 and F12 functions are plotted in Figure 5.

According to the diversity plots for all of the functions analyzed GSLEO-AEFA produces higher diversity than AEFA at each time when it is analyzed. This is an indication of well applied and well suited principles that GSLEO-AEFA contains. Gaussian mutation with local escaping operator successfully enhance the diversification ability of the original algorithm and this is well observed in the plots.

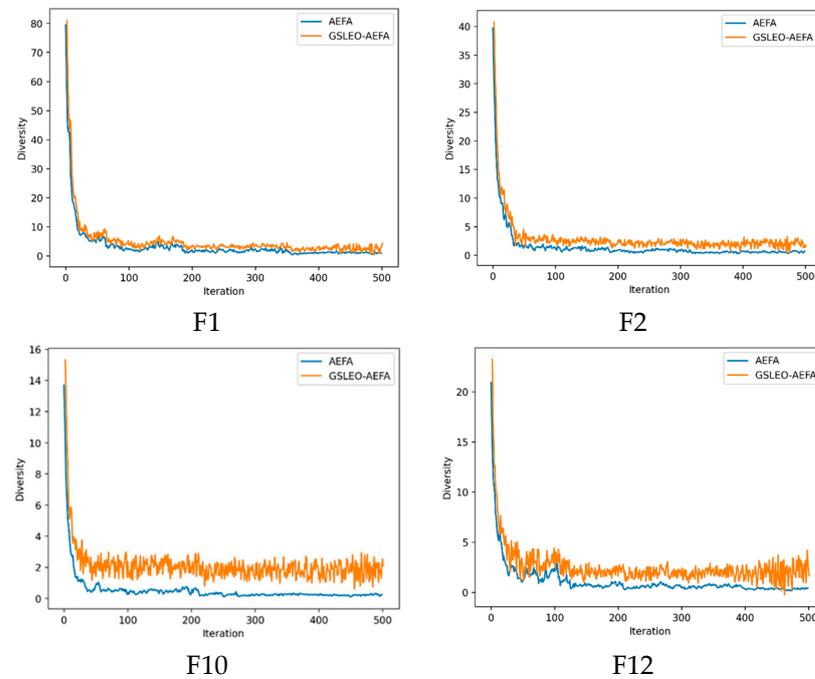
On the other hand, AEFA produces lower diversity and as a result its convergence speed is high. This was expected because of the hybrid nature of the GSLEO-AEFA, since it contains multiple methods to be executed during the optimization.

### 3.5. Computational Time and Space Complexity Analyses

The total amount of memory used by an algorithm points out the space complexity of that algorithm through the optimization process. The Big O notation of GSLEO-AEFA space complexity can be considered as  $O(\text{Max}_{\text{iteration}} \times N \times D)$  where  $\text{Max}_{\text{iteration}}$  is the number of iterations,  $N$  is the number of agents in the population and  $D$  is the dimension.

With respect to the computational time, GSLEO-AEFA and the counterpart algorithms are analyzed and the results are shown in Table 8. Out of 23 benchmark functions analyzed,

for 22 functions, GSLEO-AEFA needs higher CPU time than the others. As a consequence of its hybrid nature, the algorithm GSLEO-AEFA requires more time to figure out the optimization. On the other hand, in terms of precise solutions, GSLEO-AEFA produces high quality results and a researcher may negotiate between the high solution quality and the computational time depending on the problem specifications.



**Figure 5.** Diversity plots for GSLEO-AEFA and AEFA for the unimodal functions F1, F2 and the multimodal functions F10, F12.

**Table 8.** Computational time comparison, high values are in bold.

	GSLEO-AEFA	AEFA	PSO	DE	JAYA	SCA	CS
F1	<b>23.990</b>	23.874	40.696	1.801	10.423	10.757	9.059
F2	23.468	<b>23.744</b>	37.481	1.889	10.821	11.391	9.719
F3	<b>30.354</b>	26.628	14.308	4.643	7.535	14.579	15.415
F4	<b>25.580</b>	23.877	11.405	1.593	2.984	10.922	9.229
F5	<b>25.433</b>	24.087	11.716	1.926	3.235	11.538	9.476
F6	<b>32.218</b>	23.863	11.717	1.811	3.135	11.492	9.388
F7	<b>26.256</b>	24.369	12.371	2.109	3.374	11.440	9.427
F8	<b>25.878</b>	24.007	12.553	1.805	3.158	11.541	9.496
F9	<b>25.843</b>	24.081	11.327	1.857	3.075	10.961	9.703
F10	<b>54.693</b>	24.279	12.558	2.208	3.476	11.874	33.917
F11	<b>92.284</b>	24.405	12.796	2.244	3.545	21.765	36.307
F12	<b>90.376</b>	25.107	13.412	3.071	4.267	12.985	37.827
F13	<b>28.186</b>	25.230	13.675	3.263	4.528	12.889	11.843
F14	<b>27.039</b>	15.286	12.312	12.026	10.989	12.206	21.921
F15	<b>8.413</b>	6.382	2.331	1.406	0.968	2.211	2.726
F16	<b>6.080</b>	4.495	0.983	0.740	0.312	0.911	1.224
F17	<b>6.240</b>	4.504	1.017	0.762	0.331	0.947	1.290
F18	<b>6.562</b>	4.608	1.084	0.856	0.414	1.027	1.444
F19	<b>9.443</b>	6.186	2.421	1.785	1.328	2.296	3.366
F20	<b>11.910</b>	8.449	3.723	1.913	1.719	3.529	4.399

**Table 8.** *Cont.*

	GSLEO-AEFA	AEFA	PSO	DE	JAYA	SCA	CS
F21	<b>17.979</b>	10.459	6.707	5.602	5.251	6.713	10.894
F22	<b>21.677</b>	12.070	8.376	7.261	6.798	8.481	16.520
F23	<b>27.239</b>	14.630	24.862	9.742	9.256	11.081	21.001

### 3.6. Overall Discussion

To observe the performance of the newly proposed algorithm GSLEO-AEFA, proper analyses and a set of various experiments were conducted. Considering the original AEFA suffers from the insufficient diversification such powerful tools are added to overcome this issue as well as by allowing the algorithm to escape from local optima and to find solutions that are closer to the global optimum. By achieving this equilibrium between exploration and exploitation, high solution quality is achieved. However, there may be some limitations to consider. Since the proposed algorithm is a hybrid one and formed by multiple methods, the computational time of the algorithm is slightly worse than the other algorithms analyzed. It is because of the running each of the methods during the optimization process, respectively. However, for the ones who consider the precise solution quality, this limitation might be omitted. It is important to consider these factors carefully and to tailor the used of these techniques to the specific problems being solved.

### 3.7. Engineering Problems

The application of metaheuristics to the solution of engineering problems is one important area of research. In this section, the ability of GSLEO-AEFA to solve engineering problems was evaluated. Four common engineering problems with low and high variables from CLEC 2011 Real World Optimization Problems were used for this test [54]. The evaluation is based on 30 independent runs with 30 populations and a maximum of 1000 iterations. For comparison, the best solution from each optimization algorithm was used.

#### 3.7.1. Parameter Estimation for Frequency-Modulated (FM) Sound Waves

Frequency modulation is a complex multimodal problem. The parameter vector is defined as at  $X = [a_1, \omega_1, a_2, \omega_2, a_3, \omega_3]$  and the fitness function used in the analysis is the sum of the square errors between the estimated wave and the target wave and is given as follows:

$$F(\vec{X}) = \sum_{t=0}^{100} (y(t) - y_0(t))^2 \quad (37)$$

where

$$y(t) = a_1 \cdot \sin(\omega_1 \cdot t \cdot \theta + a_2 \cdot \sin(\omega_2 \cdot t \cdot \theta + a_3 \cdot \sin(\omega_3 \cdot t \cdot \theta))) \quad (38)$$

$$y_0(t) = (1.0) \cdot \sin((5.0) \cdot t \cdot \theta - (1.5) \cdot \sin((4.8) \cdot t \cdot \theta + (2.0) \cdot \sin((4.9) \cdot t \cdot \theta))) \quad (39)$$

The minimum value that needs to be attained is 0. The parameters in the aforementioned equation are defined in the range  $[-6.4, 6.35]$  and  $\theta = 2\pi/100$ .

Table 9 shows that GSLEO-AEFA discovered superior solution and PSO and DE followed with competitive accuracy. In comparison to other optimizers, GSLEO-AEFA has performed exceptionally well.

#### 3.7.2. Lennard–Jones Potential Problem (LJ)

The Lennard–Jones potential energy minimization problem is a multi-modal optimization problem that involves finding the minimum molecular potential energy of a molecular cluster using the Lennard–Jones potential. The problem has an exponential number of local minima, and the global minima often have structures based on Mackay icosahedrons. The Lennard–Jones potential for  $N$  atoms is given by Equation (40), and the optimization algorithm can be used to conform the molecular structure in a way that minimizes energy.

To reduce the dimensionality of the problem, one approach is to fix an atom at the origin, choose the second atom to lie on the positive X-axis, and the third atom to lie in the upper half of the X-axis. For each additional atom, three variables (coordinates of the position of the atom) are added to determine the potential energy of the cluster. The variables of the problem have three components for three atoms, six components for four atoms, and so on. The coordinates of each atom are restricted to certain ranges.

**Table 9.** Results of parameter estimation for frequency-modulated sound waves.

	GSLEO-AEFA	AEFA	PSO	DE	JAYA	SCA	CS
x(1)	1.00	0.43	−1.00	1.00	−0.63	−1.06	−1.04
x(2)	5.00	0.15	−5.00	5.00	0.00	0.00	−5.02
x(3)	−1.50	2.03	1.50	1.50	4.29	0.69	−1.36
x(4)	4.80	4.98	4.80	−4.80	4.87	0.00	−4.75
x(5)	2.00	0.83	−2.00	−2.00	0.18	−4.45	2.01
x(6)	4.90	−4.41	−4.90	4.90	0.08	−4.88	−4.90
f(x)	0	$1.963 \times 10^1$	$1.950 \times 10^{-27}$	$7.796 \times 10^{-21}$	$1.164 \times 10^1$	$1.220 \times 10^1$	1.888

Lennard–Jones pair potential for N atoms is given as follows:

$$\vec{p}_i = \{ \vec{x}_i, \vec{y}_i, \vec{z}_i \}, i = 1, \dots, N \tag{40}$$

$$V_N(p) = \sum_{i=1}^{N-1} \sum_{j=i+1}^N (r_{ij}^{-12} - 2 \cdot r_{ij}^{-6}) \tag{41}$$

where  $r_{ij} = \|\vec{p}_j - \vec{p}_i\|_2$  with gradient

$$\nabla_j V_N(p) = -12 \sum_{i=1, i \neq j}^N (r_{ij}^{-14} - r_{ij}^{-8}) (\vec{p}_j - \vec{p}_i) \tag{42}$$

The first decision variable due to the second atom, i.e.,  $x_1 \in [0, 4]$ , then the second and third variables are such that  $x_2 \in [0, 4]$  and  $x_3 \in [0, \pi]$ . The coordinates  $x_i$  for any other atom is taken to be bound in the range;  $[-4 - \frac{1}{4} [\frac{i-4}{3}], 4 + \frac{1}{4} \frac{i-4}{3}]$ .

The performance results of GSLEO-AEFA and other optimization methods in resolving the Lennard–Jones potential problem are shown in Table 10. The findings make it simple to draw the conclusion that GSLEO-AEFA has superior performance when compared to other algorithms and is capable of solving this issue with high-quality outcomes.

**Table 10.** Results Lennard–Jones Potential Problem.

	GSLEO-AEFA	AEFA	PSO	DE	JAYA	SCA	CS
x(1)	0.673	1.338	0.000	0.319	0.551	0.000	2.140
x(2)	0.329	3.307	4.000	0.567	0.035	0.000	2.789
x(3)	0.847	2.115	3.142	1.947	0.745	0.000	1.473
x(4)	0.316	3.639	−4.000	−3.980	−0.237	1.184	0.508
x(5)	−0.608	−3.723	−0.687	3.761	0.661	0.008	−0.682
x(6)	0.884	0.427	4.000	3.855	0.820	0.341	0.722
x(7)	−0.292	0.706	−1.507	−3.970	−0.992	0.337	0.393
x(8)	−0.419	1.807	−3.925	2.768	1.094	−1.074	0.265
x(9)	−0.750	−3.380	−0.706	4.250	0.559	0.005	0.443
x(10)	−0.533	−1.841	−1.260	−4.250	−0.219	−0.909	−1.221
x(11)	0.298	−2.964	−4.109	4.500	0.658	0.382	0.694
x(12)	−0.148	−3.440	0.286	4.453	1.910	−4.500	−1.556
x(13)	0.359	2.205	−1.105	4.500	2.766	0.220	0.327
x(14)	−0.092	2.806	−4.750	−3.053	0.004	0.575	−0.435
x(15)	−0.019	1.921	−1.239	−1.506	−4.750	0.921	−0.238
x(16)	−0.390	2.510	−3.925	−4.357	−0.384	0.172	−2.438
x(17)	−0.681	2.646	−0.770	3.520	0.064	−0.396	0.519

Table 10. Cont.

	GSLEO-AEFA	AEFA	PSO	DE	JAYA	SCA	CS
x(18)	0.193	−2.276	5.000	5.000	0.109	0.901	−2.169
x(19)	0.601	0.835	−1.502	−4.099	0.002	0.314	0.090
x(20)	−0.780	3.274	−4.848	5.202	−0.400	−4.879	−1.240
x(21)	−0.879	1.261	−0.337	5.188	1.740	0.073	1.516
x(22)	0.596	−0.687	−3.449	−5.197	0.242	−0.085	0.888
x(23)	−1.042	−0.199	−1.414	4.858	0.795	−0.164	−1.337
x(24)	0.065	3.430	4.400	4.681	−0.010	−0.999	0.042
x(25)	−0.130	0.348	−0.642	−4.883	1.285	−1.179	−1.877
x(26)	−1.369	3.231	−4.341	3.459	0.341	0.350	−0.091
x(27)	−0.494	2.142	−0.456	4.257	0.277	0.044	−1.566
x(28)	−0.289	0.692	−0.814	−3.372	−0.743	−0.659	1.224
x(29)	0.147	4.094	−3.390	5.390	0.003	−0.579	−0.373
x(30)	0.774	1.788	−0.224	4.311	1.361	0.296	0.015
f(x)	−26.027	−7.072	−15.126	−10.615	−14.233	−11.513	−11.646

### 3.7.3. Tersoff Potential Function Minimization Problem (TPF)

Researchers have shown a lot of interest in the evaluation of inter atomic potentials for covalent systems, notably for silicon. One such potential is the Tersoff potential, which controls the interaction of silicon atoms with strong covalent bonding [54]. Two parameterizations of silicon have been provided by Tersoff and are known as Sc (B) and Sc (C). According to the Tersoff formulation, the binding energy is expressed as a sum over atomic sites in the form,

$$E_i = \frac{1}{2} \sum_{j \neq i} f_c(r_{ij}) (V_R(r_{ij}) - B_{ij}V_A(r_{ij})) \tag{43}$$

where  $V_R$  is a repulsion term,  $V_A$  is an attraction term,  $f_c(r_{ij})$  is a switching function, and  $B_{ij}$  is a many-body term that relies on the locations of atoms  $i$  and  $j$  as well as atom  $i$ 's neighbors.

$$B_{ij} = \left(1 + \gamma^{n_1} \zeta_{ij}^{n_1}\right)^{-1/2n_1} \tag{44}$$

$n_1$  and  $\gamma$  are standard fitted parameters. For  $i$  and  $j$  the term  $\zeta_{ij}^{n_1}$  is given as,

$$\zeta_{ij} = \sum_{k \neq i} f_c(r_{ij}) g(\theta_{ijk}) \exp\left(\lambda_3^3 (r_{ij} - r_{ik})^3\right) \tag{45}$$

The bond angle between  $ij$  and  $ik$  is denoted as  $\theta_{ijk}$  which is given as,

$$g(\theta_{ijk}) = 1 + c^2/d^2 - c^2 / \left(d^2 + (h - \cos(\theta_{ijk}))^2\right) \tag{46}$$

$\lambda_3$ ,  $c$ ,  $h$  and  $d$  are known as fitted terms. From Equation (41),  $V_R$  and  $V_A$  are expressed as,

$$V_R(r_{ij}) = Ae^{-\lambda_1 r_{ij}} \tag{47}$$

$$V_A(r_{ij}) = Be^{-\lambda_2 r_{ij}} \tag{48}$$

Expressed as fitted parameters are  $A$ ,  $B$ ,  $\lambda_1$  and  $\lambda_2$ . In Equation (43), the switching function  $f_c(r_{ij})$  is expressed as,

$$f_c(r_{ij}) = \begin{cases} 1 & \text{if } r_{ij} < R - D \\ \frac{1}{2} - \frac{1}{2} \sin\left[\frac{\pi(r_{ij}-R)}{D}\right] & \text{if } R - D < r_{ij} < R + D \\ 0 & \text{if } r_{ij} \geq R + D \end{cases} \tag{49}$$

We must determine the energy of each atom in order to determine the minimal potential energy for atoms. Each atom, let us say atom  $i$  has a unique potential energy,  $E_i$ ,

which is determined by Equation (43). Therefore, in order to compute Equation (43), which requires the computation of Equations (44)–(49) for each neighbor of that atom, to know the potential energy of a single atom.

Similar to the Lennard–Jones potential problem, the first decision variable due to the second atom, i.e.,  $x_1 \in [0, 4]$ , then the second and third variables are such that  $x_2 \in [0, 4]$  and  $x_3 \in [0, \pi]$ . The coordinates  $x_i$  for any other atom is taken to be bound in the range;  $\left[-4 - \frac{1}{4} \left[\frac{i-4}{3}\right], 4 + \frac{1}{4} \frac{i-4}{3}\right]$ .

Table 11 of the test results for the TPF demonstrates that GSLEO-AEFA outperformed other optimizers and was able to offer a significantly superior solution. Additionally, it has performed effectively as engineering issues have become more constrained.

**Table 11.** Results of Tersoff Potential Function Minimization Problem.

	GSLEO-AEFA	AEFA	PSO	DE	JAYA	SCA	CS
x(1)	0.000	2.497	1.984	3.427	1.135	4.000	1.613
x(2)	4.000	1.998	2.758	3.992	3.908	0.000	0.041
x(3)	3.142	2.318	2.274	0.482	0.544	0.000	1.485
x(4)	−1.000	3.773	−0.367	−1.000	−1.000	−0.643	2.922
x(5)	−1.000	−0.847	2.762	−0.564	2.594	1.112	1.791
x(6)	3.708	2.595	2.675	1.612	1.478	0.203	3.856
x(7)	2.029	1.302	1.293	−0.282	0.650	−0.004	2.913
x(8)	3.931	4.107	−0.081	4.250	−0.336	4.250	4.246
x(9)	4.250	1.767	3.521	−1.000	1.907	0.869	−0.678
x(10)	4.250	1.168	2.749	4.250	3.417	0.378	2.498
x(11)	0.988	3.042	4.337	2.937	−0.068	−1.000	1.256
x(12)	−1.000	−0.259	0.648	4.500	0.543	−0.008	−0.216
x(13)	−0.252	3.947	2.291	4.500	0.733	2.123	2.467
x(14)	0.925	−0.869	−0.738	0.681	3.206	4.536	3.973
x(15)	4.750	0.265	1.450	3.802	2.760	0.635	3.515
x(16)	1.140	3.452	3.454	0.258	4.672	1.995	0.205
x(17)	−0.914	3.721	−1.000	1.553	−1.000	0.314	0.707
x(18)	4.581	1.334	3.504	2.542	−1.000	1.082	−0.223
x(19)	5.000	2.934	0.756	1.062	−0.443	0.957	1.043
x(20)	−1.000	4.288	4.810	0.250	5.148	0.865	2.234
x(21)	−0.085	−0.324	1.972	1.091	2.056	4.665	1.313
x(22)	0.438	1.828	4.238	−1.000	5.167	1.721	0.115
x(23)	5.500	−0.530	0.585	1.432	1.281	3.192	−0.580
x(24)	4.847	−0.955	1.934	0.299	−0.048	−0.799	4.081
x(25)	1.548	5.123	3.034	3.795	−0.393	−0.054	3.977
x(26)	5.689	1.252	4.830	2.514	−1.000	−0.474	2.986
x(27)	2.827	2.750	2.998	2.063	−0.757	2.376	1.977
x(28)	−0.686	4.798	1.078	1.325	5.491	−1.000	4.752
x(29)	−0.972	4.188	3.857	2.934	−0.865	−0.077	3.163
x(30)	6.000	4.687	4.240	−1.000	1.094	4.160	4.110
f(x)	−34.108	−14.888	−30.559	−24.606	−31.065	−20.620	−31.737

### 3.7.4. Antenna Optimization Problem

Zhang et al. suggested a validated a benchmark test suite for Antenna S-parameter optimization [55]. In the test suite there are unimodal, multimodal, and compositional functions. Each of these functions matches a typical landscape of electromagnetic simulated antenna problems. For this analysis two functions were chosen which can approximate the s-parameter optimization function behavior of two different antennas. Equations (50) and (51) model the single antenna and multi-antenna problems, respectively. Function in Equation (50) is continuous and separable and the landscape has a rose shape which is similar to the patterns of the dual band antenna. Function in Equation (51) is continuous

non-differentiable and non-separable and its landscape is similar to the reflection pattern of the (multiple input–multiple output) MIMO antenna.

$$s_1(x) = 20 \log \left( 2 \left( \sum_{i=1}^n \left| \left( \sin \left( \frac{x_i}{8} \right) \right)^2 \right| + \prod_{i=1}^n \left| \sin \left( \frac{x_i}{8} \right) \right| \right) + 1 \right) \tag{50}$$

$$s_2(x) = 100 \sqrt{|x_2 + 1 - 0.01(x_1 - 10)^2|} + 0.01|x_1| + 20 \log \left( 0.01 \left( \sum_{i=1}^n |x_i| \right)^2 (\sin(0.8x_1) + 2)^4 + 1 \right) \tag{51}$$

We evaluated GSLEO-AEFA and its counterpart algorithms with a population size of 30, iteration of 500 and a dimension set to 8, as recommended by Zhang et al. [55], by considering that the s-parameters of most antennas are around 10. The performance of each algorithm after 30 independent runs in Table 11, using average and standard deviation metrics are recorded. Our analysis indicates that GSLEO-AEFA provided the best results for both single-antenna and multi-antenna problems by demonstrating its effectiveness in solving complex optimization problems in this domain, as seen in Table 12.

**Table 12.** Results of antenna S-parameter optimization problem, significant values are in bold.

		GSLEO-AEFA	AEFA	PSO	DE	JAYA	SCA	CS
s1	AVG	<b>0.000</b>	1.351	$1.083 \times 10^{-2}$	1.603	5.448	$1.370 \times 10^1$	5.477
	STD	<b>0.000</b>	$2.469 \times 10^{-1}$	$5.946 \times 10^{-4}$	$2.752 \times 10^{-1}$	4.379	1.915	$5.862 \times 10^{-1}$
s2	AVG	<b><math>8.611 \times 10^{-7}</math></b>	$6.839 \times 10^1$	$7.255 \times 10^1$	$3.407 \times 10^1$	$9.047 \times 10^1$	$8.684 \times 10^1$	$8.040 \times 10^1$
	STD	<b><math>1.756 \times 10^{-6}</math></b>	9.564	4.589	5.161	$1.687 \times 10^{+1}$	$1.192 \times 10^1$	3.057

To observe the performance of the newly proposed algorithm GSLEO-AEFA, proper analyses and a set of various experiments were conducted. Considering the original AEFA suffers from the insufficient diversification such powerful tools are added to overcome this issue as well as by allowing the algorithm to escape from local optima and to find solutions that are closer to the global optimum. By achieving this equilibrium between exploration and exploitation, high solution quality is achieved. However, there may be some limitations to consider. Since the proposed algorithm is a hybrid one and formed by multiple methods, the computational time of the algorithm is slightly worse than the other algorithms analyzed. It is because of running each of the methods during the optimization process, respectively. However, for the ones who consider the precise solution quality, this limitation might be omitted. It is important to consider these factors carefully and to tailor the used of these techniques to the specific problems being solved.

#### 4. Conclusions

The performance of the proposed methodology GSLEO-AEFA with its counterpart algorithms were tested on 23 benchmark functions by conducting overall effectiveness, convergence analysis, Friedman and Wilcoxon rank-sum statistical tests as well as computational analysis. Even in the varying dimension sizes, GSLEO-AEFA has achieved the best optimization accuracy among the algorithms tested.

The performance of GSLEO-AEFA was then validated on real engineering design problems including antenna optimization. In here, it has been outlined that the success of the proposed algorithm was maintained at the same level for all of the design problems. The proper use of local escaping operator (LEO), Gaussian mutation (GM) and specular reflection learning (SRL) techniques in the algorithm GSLEO-AEFA provides a better balance in exploration and exploitation by overcoming not only the original algorithm which is derived from but the other algorithms used in the comparison.

Although its performance is quite satisfactory, it can be acknowledged that the computational time is not always satisfactory. This is mainly caused because of the structure

of the GSLEO-AEFA, since it was generated by having three individual methods as well as AEFA. However, researchers may debate for a fine balance between the optimization accuracy and the computational time for the problem to be solved.

**Author Contributions:** O.R.A.; designing experiment, doing tests, data analysis and writing. E.D.Ü.; project administration, data analysis, writing, revising and editing. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** The study was conducted according to the guidelines of the Declaration of Helsinki. The approval of the Institutional Review Board was exempted due to the use of publicly available data.

**Informed Consent Statement:** Patient consent for the use of anonymized biological material was in accordance with Article 34 of the Human Research Act, The Federal Assembly of the Swiss Confederation.

**Data Availability Statement:** The data obtained through the experiments are available upon a reasonable request from the first author, O.R.A.

**Conflicts of Interest:** The authors declare that there are no conflict of interest.

## References

1. Mahariq, I. On the application of the spectral element method in electromagnetic problems involving domain decomposition. *Turk. J. Electr. Eng. Comput. Sci.* **2017**, *25*, 1059–1069. [[CrossRef](#)]
2. Yang, C.X.; Zhang, J.; Tong, M.S. A Hybrid Quantum-Behaved Particle Swarm Optimization Algorithm for Solving Inverse Scattering Problems. *IEEE Trans. Antennas Propag.* **2021**, *69*, 5861–5869. [[CrossRef](#)]
3. Donelli, M.; Franceschini, D.; Rocca, P.; Massa, A. Three-Dimensional Microwave Imaging Problems Solved Through an Efficient Multiscaling Particle Swarm Optimization. *IEEE Trans. Geosci. Remote Sens.* **2009**, *47*, 1467–1481. [[CrossRef](#)]
4. Gupta, S.; Deep, K. A memory-based Grey Wolf Optimizer for global optimization tasks. *Appl. Soft Comput.* **2020**, *93*, 106367. [[CrossRef](#)]
5. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95-International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
6. Abbas, M.; Alshehri, M.A.; Barnawi, A.B. Potential Contribution of the Grey Wolf Optimization Algorithm in Reducing Active Power Losses in Electrical Power Systems. *Appl. Sci.* **2022**, *12*, 6177. [[CrossRef](#)]
7. El Gmili, N.; Mjahed, M.; El Kari, A.; Ayad, H. Particle Swarm Optimization and Cuckoo Search-Based Approaches for Quadrotor Control and Trajectory Tracking. *Appl. Sci.* **2019**, *9*, 1719. [[CrossRef](#)]
8. Li, Y.; Pei, Y.-H.; Liu, J.-S. Bat optimal algorithm combined uniform mutation with Gaussian mutation. *Kongzhi Yu Juece/Control Decis.* **2017**, *32*, 1775–1781. [[CrossRef](#)]
9. Yuan, X.; Miao, Z.; Liu, Z.; Yan, Z.; Zhou, F. Multi-Strategy Ensemble Whale Optimization Algorithm and Its Application to Analog Circuits Intelligent Fault Diagnosis. *Appl. Sci.* **2020**, *10*, 3667. [[CrossRef](#)]
10. Ni, J.; Tang, J.; Wang, R. Hybrid Algorithm of Improved Beetle Antenna Search and Artificial Fish Swarm. *Appl. Sci.* **2022**, *12*, 13044. [[CrossRef](#)]
11. Sun, L.; Feng, B.; Chen, T.; Zhao, D.; Xin, Y. Equalized Grey Wolf Optimizer with Refraction Opposite Learning. *Comput. Intell. Neurosci.* **2022**, *2022*, e2721490. [[CrossRef](#)]
12. David, R.-C.; Precup, R.-E.; Petriu, E.M.; Rădac, M.-B.; Preitl, S. Gravitational search algorithm-based design of fuzzy control systems with a reduced parametric sensitivity. *Inf. Sci.* **2013**, *247*, 154–173. [[CrossRef](#)]
13. Mirjalili, S. SCA: A Sine Cosine Algorithm for solving optimization problems. *Knowl.-Based Syst.* **2016**, *96*, 120–133. [[CrossRef](#)]
14. Kaveh, A.; Mahdavi, V.R. Colliding Bodies Optimization method for optimum discrete design of truss structures. *Comput. Struct.* **2014**, *139*, 43–53. [[CrossRef](#)]
15. Zheng, H.; Gao, J.; Xiong, J.; Yao, G.; Cui, H.; Zhang, L. An Enhanced Artificial Electric Field Algorithm with Sine Cosine Mechanism for Logistics Distribution Vehicle Routing. *Appl. Sci.* **2022**, *12*, 6240. [[CrossRef](#)]
16. Juárez-Pérez, F.; Cruz-Chávez, M.A.; Rivera-López, R.; Ávila-Melgar, E.Y.; Eraña-Díaz, M.L.; Cruz-Rosales, M.H. Grid-Based Hybrid Genetic Approach to Relaxed Flexible Flow Shop with Sequence-Dependent Setup Times. *Appl. Sci.* **2022**, *12*, 607. [[CrossRef](#)]
17. Wang, S.L.; Adnan, S.H.; Ibrahim, H.; Ng, T.F.; Rajendran, P. A Hybrid of Fully Informed Particle Swarm and Self-Adaptive Differential Evolution for Global Optimization. *Appl. Sci.* **2022**, *12*, 11367. [[CrossRef](#)]
18. Eltaieb, T.; Mahmood, A. Differential Evolution: A Survey and Analysis. *Appl. Sci.* **2018**, *8*, 1945. [[CrossRef](#)]
19. Fogel, G.B. Evolutionary Programming. In *Handbook of Natural Computing*; Rozenberg, G., Bäck, T., Kok, J.N., Eds.; Springer: Berlin, Heidelberg, 2012; pp. 699–708, ISBN 978-3-540-92910-9.

20. Liu, X.; Wang, Y.; Zhou, M. Dimensional Learning Strategy-Based Grey Wolf Optimizer for Solving the Global Optimization Problem. *Comput. Intell. Neurosci.* **2022**, *2022*, 3603607. [[CrossRef](#)]
21. Črepinšek, M.; Liu, S.-H.; Mernik, M. Exploration and exploitation in evolutionary algorithms: A survey. *ACM Comput. Surv.* **2013**, *45*, 35:1–35:33. [[CrossRef](#)]
22. Anita; Yadav, A. AEFA: Artificial electric field algorithm for global optimization. *Swarm Evol. Comput.* **2019**, *48*, 93–108. [[CrossRef](#)]
23. Sinthia, P.; Malathi, M. Cancer detection using convolutional neural network optimized by multistrategy artificial electric field algorithm. *Int. J. Imaging Syst. Technol.* **2021**, *31*, 1386–1403. [[CrossRef](#)]
24. Petwal, H.; Rani, R. An Improved Artificial Electric Field Algorithm for Multi-Objective Optimization. *Processes* **2020**, *8*, 584. [[CrossRef](#)]
25. Selem, S.I.; El-Fergany, A.A.; Hasanien, H.M. Artificial electric field algorithm to extract nine parameters of triple-diode photovoltaic model. *Int. J. Energy Res.* **2021**, *45*, 590–604. [[CrossRef](#)]
26. Niroomand, S. Hybrid artificial electric field algorithm for assembly line balancing problem with equipment model selection possibility. *Knowl.-Based Syst.* **2021**, *219*, 106905. [[CrossRef](#)]
27. Das, H.; Naik, B.; Behera, H.S. Optimal Selection of Features Using Artificial Electric Field Algorithm for Classification. *Arab. J. Sci. Eng.* **2021**, *46*, 8355–8369. [[CrossRef](#)]
28. Houssein, E.H.; Hashim, F.A.; Ferahtia, S.; Rezk, H. An efficient modified artificial electric field algorithm for solving optimization problems and parameter estimation of fuel cell. *Int. J. Energy Res.* **2021**, *45*, 20199–20218. [[CrossRef](#)]
29. Bi, J.; Zhou, Y.; Tang, Z.; Luo, Q. Artificial electric field algorithm with inertia and repulsion for spherical minimum spanning tree. *Appl. Intell.* **2022**, *52*, 195–214. [[CrossRef](#)]
30. Alanazi, A.; Alanazi, M. Artificial Electric Field Algorithm-Pattern Search for Many-Criteria Networks Reconfiguration Considering Power Quality and Energy Not Supplied. *Energies* **2022**, *15*, 5269. [[CrossRef](#)]
31. Cheng, J.; Xu, P.; Xiong, Y. An improved artificial electric field algorithm and its application in neural network optimization. *Comput. Electr. Eng.* **2022**, *101*, 108111. [[CrossRef](#)]
32. Malisetti, N.; Pamula, V.K. Energy efficient cluster based routing for wireless sensor networks using moth levy adopted artificial electric field algorithm and customized grey wolf optimization algorithm. *Microprocess. Microsyst.* **2022**, *93*, 104593. [[CrossRef](#)]
33. Nayak, S.C.; Sanjeev Kumar Dash, C.; Behera, A.K.; Dehuri, S. An Elitist Artificial-Electric-Field-Algorithm-Based Artificial Neural Network for Financial Time Series Forecasting. In *Biologically Inspired Techniques in Many Criteria Decision Making*; Dehuri, S., Prasad Mishra, B.S., Mallick, P.K., Cho, S.-B., Eds.; Springer Nature: Singapore, 2022; pp. 29–38.
34. Rahnamayan, S.; Tizhoosh, H.R.; Salama, M.M.A. Opposition-Based Differential Evolution. *IEEE Trans. Evol. Comput.* **2008**, *12*, 64–79. [[CrossRef](#)]
35. Zhang, Y. Backtracking search algorithm with specular reflection learning for global optimization. *Knowl.-Based Syst.* **2021**, *212*, 106546. [[CrossRef](#)]
36. Alkayem, N.F.; Shen, L.; Al-hababi, T.; Qian, X.; Cao, M. Inverse Analysis of Structural Damage Based on the Modal Kinetic and Strain Energies with the Novel Oppositional Unified Particle Swarm Gradient-Based Optimizer. *Appl. Sci.* **2022**, *12*, 11689. [[CrossRef](#)]
37. Alkayem, N.F.; Shen, L.; Asteris, P.G.; Sokol, M.; Xin, Z.; Cao, M. A new self-adaptive quasi-oppositional stochastic fractal search for the inverse problem of structural damage assessment. *Alex. Eng. J.* **2022**, *61*, 1922–1936. [[CrossRef](#)]
38. Adegboye, O.R.; Ülker, E.D. A Quick Performance Assessment for Artificial Electric Field Algorithm. In Proceedings of the 2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Ankara, Turkey, 9–11 June 2022; pp. 1–5.
39. Sajwan, A.; Yadav, A. A study of exploratory and stability analysis of artificial electric field algorithm. *Appl. Intell.* **2022**, *52*, 10805–10828. [[CrossRef](#)]
40. Yang, W.; Xia, K.; Li, T.; Xie, M.; Zhao, Y. An Improved Transient Search Optimization with Neighborhood Dimensional Learning for Global Optimization Problems. *Symmetry* **2021**, *13*, 244. [[CrossRef](#)]
41. Liu, Y.; Heidari, A.A.; Cai, Z.; Liang, G.; Chen, H.; Pan, Z.; Alsufyani, A.; Bourouis, S. Simulated annealing-based dynamic step shuffled frog leaping algorithm: Optimal performance design and feature selection. *Neurocomputing* **2022**, *503*, 325–362. [[CrossRef](#)]
42. Dai, M.; Feng, X.; Yu, H.; Guo, W. An opposition-based differential evolution clustering algorithm for emotional preference and migratory behavior optimization. *Knowl.-Based Syst.* **2023**, *259*, 110073. [[CrossRef](#)]
43. Nama, S.; Saha, A.K.; Sharma, S. Performance up-gradation of Symbiotic Organisms Search by Backtracking Search Algorithm. *J. Ambient. Intell. Hum. Comput.* **2022**, *13*, 5505–5546. [[CrossRef](#)]
44. Duan, Y.; Liu, C.; Li, S.; Guo, X.; Yang, C. Gaussian Perturbation Specular Reflection Learning and Golden-Sine-Mechanism-Based Elephant Herding Optimization for Global Optimization Problems. *Comput. Intell. Neurosci.* **2021**, *2021*, e9922192. [[CrossRef](#)]
45. Emambocus, B.A.S.; Jasser, M.B.; Mustapha, A.; Amphawan, A. Dragonfly Algorithm and Its Hybrids: A Survey on Performance, Objectives and Applications. *Sensors* **2021**, *21*, 7542. [[CrossRef](#)]
46. Ahmadianfar, I.; Bozorg-Haddad, O.; Chu, X. Gradient-based optimizer: A new metaheuristic optimization algorithm. *Inf. Sci.* **2020**, *540*, 131–159. [[CrossRef](#)]
47. Storn, R.; Price, K. Differential Evolution—A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [[CrossRef](#)]

48. Venkata Rao, R. Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. *Int. J. Ind. Eng. Comput.* **2016**, *7*, 19–34. [[CrossRef](#)]
49. Yang, X.-S.; Deb, S. Cuckoo Search via Lévy flights. In Proceedings of the 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), Coimbatore, India, 9–11 December 2009; pp. 210–214.
50. Qaraad, M.; Amjad, S.; Hussein, N.K.; Mirjalili, S.; Halima, N.B.; Elhosseini, M.A. Comparing SSALEO as a Scalable Large Scale Global Optimization Algorithm to High-Performance Algorithms for Real-World Constrained Optimization Benchmark. *IEEE Access* **2022**, *10*, 95658–95700. [[CrossRef](#)]
51. García, S.; Molina, D.; Lozano, M.; Herrera, F. 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. *J. Heuristics* **2008**, *15*, 617. [[CrossRef](#)]
52. Derrac, J.; García, S.; Molina, D.; Herrera, F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol. Comput.* **2011**, *1*, 3–18. [[CrossRef](#)]
53. Zamani, H.; Nadimi-Shahraki, M.H.; Gandomi, A.H. CCSA: Conscious Neighborhood-based Crow Search Algorithm for Solving Global Optimization Problems. *Appl. Soft Comput.* **2019**, *85*, 105583. [[CrossRef](#)]
54. Swagatam, D.; Suganthan, P.N. *Problem Definitions and Evaluation Criteria for CEC 2011 Competition on Testing Evolutionary Algorithms on Real World Optimization Problems*; Jadavpur University, Nanyang Technological University: Kolkata, India, 2011.
55. Zhang, Z.; Chen, H.; Jiang, F.; Yu, Y.; Cheng, Q.S. A Benchmark Test Suite for Antenna S-Parameter Optimization. *IEEE Trans. Antennas Propagat.* **2021**, *69*, 6635–6650. [[CrossRef](#)]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.