Research Article

An Improved Electromagnetic Field Optimization for the Global Optimization Problems

Alkin Yurtkuran

Dept. of Industrial Engineering, Bursa Uludag University, Bursa, Turkey

Correspondence should be addressed to Alkin Yurtkuran; alkin@uludag.edu.tr

Received 21 January 2019; Revised 15 April 2019; Accepted 16 April 2019; Published 23 May 2019

Academic Editor: Juan Carlos Fernández

Copyright © 2019 Alkin Yurtkuran. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Electromagnetic field optimization (EFO) is a relatively new physics-inspired population-based metaheuristic algorithm, which simulates the behavior of electromagnets with different polarities and takes advantage of a nature-inspired ratio, known as the golden ratio. In EFO, the population consists of electromagnetic particles made of electromagnets corresponding to variables of an optimization problem and is divided into three fields: positive, negative, and neutral. In each iteration, a new electromagnetic particle is generated based on the attraction-repulsion forces among these electromagnetic fields, where the repulsion force helps particle to avoid the local optimal point, and the attraction force leads to find global optimal. This paper introduces an improved version of the EFO called improved electromagnetic field optimization (iEFO). Distinct from the EFO, the iEFO has two novel modifications: new solution generation function for the electromagnets and adaptive control of algorithmic parameters. In addition to these major improvements, the boundary control and randomization procedures for the newly generated electromagnets are modified. In the computational studies, the performance of the proposed iEFO is tested against original EFO, existing physics-inspired algorithms, and state-of-the-art meta-heuristic algorithms as artificial bee colony algorithm, particle swarm optimization, and differential evolution. Obtained results are verified with statistical testing, and results reveal that proposed iEFO outperforms the EFO and other considered competitor algorithms by providing better results.

1. Introduction

Nowadays, the use of metaheuristic algorithms has surprisingly increased for solving various problems due to their flexibility, gradient-free mechanism, and local optima avoidance structures [1]. Although they do not guarantee the optimal solution for an optimization problem, they mostly have a capability of finding the near-optimal solution. During the last few decades, various algorithms have been proposed to solve different real-life problems. In spite of the fact that most of them show similar behavior while searching the solution space, each algorithm may have superior performance on a specific problem as described in “No Free Lunch” theorem, which states that no single algorithm can perform well on every optimization problem [2].

Nature-inspired optimization algorithms have been widely adopted in the area of computer science, mathematics, control, or decision making due to their efficient

performance on solving complex optimization problems since the last few decades [3]. Considering the existing literature, different classifications for the nature-inspired metaheuristic algorithms are given based on a specific philosophy. These algorithms are simply classified into three main groups: evolution-inspired, physics-inspired, and swarm-inspired [4–6]. Among this classification, physics-inspired algorithms simulate physical laws in the universe, and they are different from other approaches because of their search agents based on physics rules [7, 8]. A considerable number of metaheuristic algorithms in the literature have taken inspiration from physical phenomena. A detailed review of the physics-inspired metaheuristic algorithms are given in the study of Can and Alatas [9] and Tahani and Babayan [10], where the list of the algorithms reviewed in these studies is presented in Table 1. Moreover, Table 1 also includes the algorithms, which are not given in both the studies. Considering the list of the physics-inspired

TABLE 1: List of the physics-inspired algorithms.

Algorithm	Reference	Year	Inspiration	Reviewed in [9]	Reviewed in [10]
Atom search algorithm	Zhao et al. [6]	2018	Basic molecular dynamics		
Flow regime algorithm	Tahani and Babayan [10]	2018	Concepts of fluid mechanics		
Hydrological cycle algorithm	Wedyan et al. [11]	2017	The continuous movement of water in nature		
Thermal exchange optimization	Kaveh and Dadras [12]	2017	Newton's law of cooling		
Weighted superposition algorithm	Baykasoğlu and Akpinar [13]	2017	Superposition principle and attraction		
Electromagnetic field optimization	Abedinpourshotorban et al. [14]	2016	Attraction-repulsion forces among electromagnets		
Rain water algorithm	Biyanto et al. [15]	2016	The pattern of physically rain water movements		
Ion motion algorithm	Javidy et al. [16]	2015	Ions motion in nature	✓	✓
Colliding bodies optimization	Kaveh and Mahdavi [17]	2014	Collision between objects		
Black hole optimization	Hatamlou [18]	2013	Black hole phenomenon		✓
Dłotko-Specogna	Dłotko and Specogna [19]	2013	Low-frequency electrodynamics		
Ray optimization	Kaveh and Khayatizad [20]	2012	The transition of ray from one medium to another from physics	✓	✓
Water cycle algorithm	Eskandar et al. [21]	2012	Water cycle process and how rivers and streams flow to the sea in the real-world	✓	✓
Galaxy based search algorithm	Shah-Hosseini [22]	2011	The arm of spiral galaxies in the outer space	✓	✓
Gravitational interactions optimization	Flores et al. [23]	2011	Gravitational forces produced by the interaction of the masses of a set of bodies	✓	✓
Spiral optimization algorithm	Tamura and Yasuda [24]	2011	The analogy of spiral phenomena in nature	✓	
Water flow algorithm	HIEU [25]	2011	Characteristics of water droplets and their erosion capability to overcome obstacles	✓	✓
Charged system search	Kaveh and Talatahari [26]	2010	Coulomb law from electrostatics and the Newtonian laws of mechanics	✓	✓
Gravitation field algorithm	Zheng et al. [27]	2010	Astronomy theory solar nebular disk model of planetary formation	✓	
Artificial physics algorithm	Xie et al. [28]	2009	Newton's second law	✓	✓
Gravitational search algorithm	Rashedi et al. [29]	2009	Law of gravity and mass interactions	✓	✓
Intelligent water drops algorithm	Shah-Hosseini [30]	2009	Natural water drops that flow in rivers	✓	✓
Light ray optimization algorithm	Shen and Li [31]	2009	Optical refraction and reflection of light rays	✓	✓
Big Crunch optimization	Kripka and Kripka [32]	2008	Cosmological theory is known as closed universe	✓	✓
Magnetic optimization algorithm	Tayarani-N and Akbarzadeh-T [33]	2008	Principles of magnetic field theory	✓	✓
Central force optimization algorithm	[34]	2007	The metaphor of gravitational kinematics	✓	✓
Integrated radiation algorithm	Chuang and Jiang [35]	2007	Gravitational radiation in the curvature of space-time	✓	✓
River formation dynamics algorithm	Rabanal et al. [36]	2007	River formation by water	✓	✓
Big Bang-Big Crunch algorithm	Erol and Eksin [37]	2006	Energy dissipation (Big Bang) and center of mass (Big Crunch)	✓	✓
Particle collision algorithm	Sacco and De Oliveira [38]	2005	Nuclear collision reactions, particularly scattering and absorption	✓	✓
Space gravitational algorithm	Hsiao et al. [39]	2005	Einstein's general theory of relativity	✓	✓
Electromagnetism-like algorithm	Birbil and Fang [40]	2003	Attraction-repulsion mechanism of the sample points	✓	✓
Hysteretic optimization	Zarand et al. [41]	2002	Demagnetization procedure	✓	✓
Harmony search	Geem et al. [42]	2001	Improvisation of the music player		✓
Simulated annealing	Kirkpatrick et al. [43]	1983	Annealing process of molten metals		✓

metaheuristic algorithms, cumulative numbers of the algorithms by the years are given in Figure 1. It should be noted from Figure 1 that most of these algorithms have been proposed in the last decade.

Regarding the physics-inspired metaheuristic algorithms given in Table 1, the electromagnetic field optimization (EFO) is one of the relatively new physics-inspired metaheuristic algorithms, which is first proposed by Abedinpourshotorban et al. [14]. The EFO is inspired by the behavior of electromagnets with different polarities and takes advantage of a nature-inspired ratio, known as the golden ratio. In EFO, the population consists of electromagnetic particles made of electromagnets corresponding to variables of the optimization problem and is divided into three fields: positive, negative, and neutral. In each iteration, a new electromagnetic particle is generated based on the attraction-repulsion forces among these electromagnetic fields, where the repulsion force helps the particle to avoid the local optimal point and the attraction force leads to find global optimal point.

Since the EFO is a recently proposed algorithm, the number of studies that consider this approach is limited. Yurtkuran and Kucukoglu [44] used the original version of the EFO and also other three different physics-inspired metaheuristic algorithms such as electromagnetism-like algorithm, gravitational search algorithm, and weighted superposition attraction algorithm for the solar cell parameter estimation problem to optimize the performance of solar systems. The authors compared the performance of the algorithms on a well-known benchmark problem set. Their computational studies show that EFO outperforms the other three algorithms and provides better results. Boucekara et al. [45] proposed a modified version of the EFO to identify optimal coordination of directional overcurrent relays for power systems protection, which is a nonlinear and highly constrained optimization problem. The authors applied two simple modifications on the algorithm by changing the uniformly distributed random generation procedure (used in the search equation) to the normal distributed random generation and boundary check procedure of the electromagnets. In computational studies, the efficiency of the modified version of the EFO is shown. Talebi and Dehkordi [46] introduced a binary version of the EFO for the sensitive association rules hiding for personal information protection. Performance of the algorithm is evaluated by doing experiments on both real-world and synthetic datasets. Better results are observed by binary EFO comparing with four different algorithms.

In general, metaheuristic algorithms are established to balance two important factors for their performance, which are exploration and exploitation. The exploration means the ability to investigate in the unvisited regions of the search space, whereas exploitation refers to the ability to search around better solutions [47]. However, the lack of the right balance between these factors leads to poor convergence when solving complex problems. Therefore, many studies have been made to improve the performance of metaheuristic algorithms based on three strategies [48]: (i) hybridization of the algorithms [49–51], (ii) adaptation

mechanism [52–54], and (iii) introducing new operators for generating new solutions [55–57]. In this manner, this paper introduces an improved version of the EFO, which is called improved electromagnetic field optimization (iEFO). Concerning the original EFO, the iEFO has two novel changes: new solution generation function for the particles and adaptive control of algorithmic parameters. With the new solution procedure, the selection probability of the better solutions in the candidate particle generation step is increased. Furthermore, by an adaptive control mechanism, it explores the search space effectively, especially in the early stages of the search process, whereas exploitation is emphasized in the latter phases. In addition to these major improvements, the boundary control and randomization procedures for the newly generated particles are modified.

A modified version of the EFO is introduced by Boucekara et al. [45]. However, the modified version of the EFO proposed by the authors includes minor changes in random number generation and boundary condition in the new electromagnet generating step. To the best of our knowledge, this paper is the first improved version of the EFO with major changes on the search mechanism, new solution generation, and control of search parameters. The main contributions of the proposed study can be summarized as follows: (i) a new search equation is defined to balance the exploration and exploitation performance, (ii) in order to further improve the exploitation behavior, the parameters of the new search equation are controlled adaptively, and (iii) a detailed literature review of physics-inspired metaheuristic algorithms is presented. In computational studies, the performance of the proposed iEFO is analyzed on a well-known global optimization benchmark problems into three cases. First, the proposed iEFO is compared with the original version of the EFO and also four recent physics-inspired population-based metaheuristic algorithms. Second, the proposed iEFO is compared with original version of the well-known artificial bee colony (ABC) algorithm [58], differential evolution (DE) algorithm [59], and particle swarm optimization (PSO) algorithm [60]. Third, the proposed iEFO is compared with the improved variants of PSO and DE, and detailed statistical analyses for the performance comparisons are given in the computational studies.

The remainder of the paper is organized as follows. In Section 2, the original version of the EFO and its main steps are described. Section 3 introduces the proposed iEFO for global optimization problems. Computational results are given in Section 4. Finally, a conclusion part of future research perspectives is provided in Section 5.

2. Electromagnetic Field Optimization (EFO)

The EFO is a relatively new metaheuristic algorithm inspired by attraction-repulsion forces among electromagnets with different polarities and nature-inspired ratio called the golden ratio [14]. In EFO, a solution is represented by electromagnetic particle (EMP) made of electromagnets, and the number of electromagnets is equal to the number of variables in the optimization problem. Different from

FIGURE 1: Cumulative numbers of the proposed physics-inspired metaheuristic algorithms by the years.

permanent magnets, each electromagnet in EMP has the same single polarity (positive or negative), and each electromagnet can apply a force of attraction or repulsion among other neighbor electromagnets. Table 2 summarizes the notations used in the EFO. In this context, the main steps of the algorithm are described in this section.

2.1. Initialization. As in most of the population-based metaheuristic algorithms, the EFO starts by randomly generating a population of electromagnetic particles by using equation (1). Each electromagnet of an EMP is randomly generated within its lower and upper bounds. After a population randomly generated, EMPs are sorted based on their fitness value in descending order:

$$\text{EMP}_j^i = L_j + \text{rand}(U_j - L_j), \quad (1)$$

$$i = 1, \dots, N_{\text{emp}}, j = 1, \dots, N_{\text{var}}.$$

2.2. Classification. In the classification phase of the EFO, the EMPs are classified into three groups with different polarities: positive field that contains the best EMPs, negative field that contains worst EMPs, and neutral field that contains small negative polarities. The number of EMPs of these three groups is determined by using two control parameters: P_{field} and N_{field} . The P_{field} and N_{field} represent the percentage of the allocated solution for positive and negative

TABLE 2: Notations used in the EFO.

N_{emp}	Number of EMPs in population
N_{var}	Number of electromagnets of EMP
EMP_j^i	Value of electromagnet j in particle i , $i = 1, \dots, N_{\text{emp}}, j = 1, \dots, N_{\text{var}}$
$\text{EMP}_j^{\text{New}}$	Value of electromagnet j in new solution, $j = 1, \dots, N_{\text{var}}$
U_j	Upper bound of the j^{th} electromagnet, $j = 1, \dots, N_{\text{var}}$
L_j	Lower bound of the j^{th} electromagnet, $j = 1, \dots, N_{\text{var}}$
P_j	Random index from positive field generated for each electromagnet for the candidate solution, $j = 1, \dots, N_{\text{var}}$
N_j	Random index from negative field generated for each electromagnet for the candidate solution, $j = 1, \dots, N_{\text{var}}$
K_j	Random index from neutral field generated for each electromagnet for the candidate solution, $j = 1, \dots, N_{\text{var}}$
P_{field}	Portion of population, which belongs to positive field
N_{field}	Portion of population, which belongs to negative field
Ps_{rate}	Probability of selecting variables of generated particle
R_{rate}	Probability of changing one variable of a generated particle
ϕ	Golden ratio
rand	Uniform random number $\text{rand} \in (0, 1)$
rand_int	Integer random number
RI	Index of one electromagnet of generated particle used for randomization step

part, respectively. The remaining solutions form the neutral part. According to the classification of the population, selection of one EMP from each field for the new electromagnet generation procedure is made by using the following equations:

$$\begin{aligned} P_j &= \text{rnd_int}(1, \lfloor N_emp * P_field \rfloor), \quad j = 1, \dots, N_var, \\ N_j &= \text{rnd_int}(\lfloor N_emp * (1 - N_field) \rfloor, N_emp), \\ &\quad j = 1, \dots, N_var, \\ K_j &= \text{rnd_int}(\lceil N_emp * P_field \rceil, \lceil N_emp * (1 - N_field) \rceil), \\ &\quad j = 1, \dots, N_var. \end{aligned} \quad (2)$$

$$\text{EMP}_j^{\text{New}} = \begin{cases} \text{EMP}_j^{P_j}, & \text{if rand} < Ps_rate, \\ \text{EMP}_j^{K_j} + (\varphi * \text{rand})(\text{EMP}_j^{P_j} - \text{EMP}_j^{K_j}) - \text{rand}(\text{EMP}_j^{N_j} - \text{EMP}_j^{K_j}), & \text{otherwise,} \end{cases} \quad (3)$$

$$j = 1, \dots, N_var.$$

After an electromagnet is generated for the new solution, its bounds are checked whether it is in the range of lower and upper limit. If the new electromagnet violates its bounds, then it is regenerated randomly within its bounds as in the initialization step.

2.4. Randomization. In each iteration, the randomization step is applied to change only one electromagnet of generated EMP with randomly generated electromagnet within its range as follows:

$$\text{EMP}_{\text{RI}}^{\text{New}} = L_{\text{RI}} + \text{rand}(U_{\text{RI}} - L_{\text{RI}}), \quad \text{if rand} < R_rate, \quad (4)$$

where RI is an integer counter used to identify the index of the electromagnet for randomization. RI is initialized with 1 or randomly generated integer number between $[1, N_var]$ at the beginning and is increased by 1 in each randomization process. In case the RI reaches to its upper limit N_var , then it is set to 1 again.

2.5. Selection. At the end of each iteration, the fitness function value of the new solution is determined. If the

2.3. New Solution Generation. New solution generation procedure is the most important step of the EFO. After specifying the randomly selected indexes from positive, neutral, and negative fields, a new electromagnet for the candidate solution is generated by using equation (5) as follows. If a uniformly distributed random number is lower than the Ps_rate , then the corresponding electromagnet of the candidate solution is set to the randomly selected electromagnet from the positive field. Otherwise, the corresponding electromagnet of the candidate solution is generated based on the randomly selected electromagnet from the neutral field, which is affected by the randomly selected electromagnets from positive and negative fields:

generated EMP provides a better solution than the worst EMP (N_emp^{th} particle) in the population, then the new solution is inserted into the sorted population according to its fitness function value and the worst EMP is removed from the population.

3. Proposed iEFO

Generally, metaheuristic algorithms try to balance the two important factors for their performance as exploration and exploitation. The candidate solution generation mechanism (search equation) plays an important role in determining the performance of a metaheuristic algorithm [61]. Up to now, researches have focused on different search equation modifications to improve the performance of well-known algorithms such as ABC, PSO, and DE. Inspired by these studies, a novel search equation is defined for iEFO in this study. As can be seen in equation (5), the original EFO uses the EMP from the neutral field and generate a new candidate solution using this EMP with positive feedback from the positive field, whereas negative feedback from the negative field. In the iEFO, a new search equation is proposed as follows:

$$\text{EMP}_j^{\text{New}} = \begin{cases} \text{EMP}_j^{P_j}, & \text{if rand} < Ps_rate, \\ \text{EMP}_j^{\text{RW}_j} + (\varphi * \text{rand})(\text{EMP}_j^{P_j} - \text{EMP}_j^{K_j}) - \text{rand}(\text{EMP}_j^{N_j} - \text{EMP}_j^{K_j}), & \text{otherwise,} \end{cases} \quad (5)$$

$$j = 1, \dots, N_var.$$

Similar to the original EFO, P_j , N_j , and K_j are the indexes of the selected EMPs from positive, negative, and

neutral parts, respectively. φ is the golden ratio constant, which is used to guide the candidate solutions towards the

positive part. Here, RW_j depicts the selected EMP using the roulette wheel method, which is different from P_j , N_j , and K_j . In other words, in the iEFO, the candidate EMP is constructed based on the EMP, which is selected from the roulette wheel. Equation (7) can generate the candidate solution not only around the neutral field but also positive and negative fields, while better EMPs have higher probability to be selected. It is clear that guiding the search by RW_j will improve the exploitation ability of EFO.

Next, to further improve the exploitation performance of iEFO, an adaptive mechanism is employed. It is well known that integrating adaptive control mechanisms into metaheuristic algorithms is a very popular technique in the related literature [52, 54, 62]. In iEFO, two main control parameters, i.e., Ps_rate and R_rate , are controlled adaptively over the course of a run. As mentioned in the previous section, Ps_rate is responsible for the probability of copying the index of the EMP from the positive field, whereas R_rate is the probability of the randomization procedure. In the iEFO, Ps_rate and R_rate are updated at the end of each iteration as follows:

$$Ps_rate = Ps_{RMin} + \frac{Iter \times (Ps_{RMax} - Ps_{RMin})}{MaxIter}, \quad (6)$$

$$R_rate = R_{RMax} - \frac{Iter \times (R_{RMax} - R_{RMin})}{MaxIter}, \quad (7)$$

where $Iter$ and $MaxIter$ refer to the current iteration value and the maximum iteration value, respectively. As can be seen from equation (6), Ps_rate is increased from Ps_{RMin} to Ps_{RMax} during the search process. Similarly, R_rate is reduced adaptively from R_{RMax} to Ps_{RMin} . Here, Ps_{RMin} , Ps_{RMax} , R_{RMax} , and R_{RMin} are the new control parameters of iEFO, which will be set before the search process. In a word, the two new control equation of Ps_rate and R_rate will improve the exploration-exploitation balance by giving a higher probability to random search mechanism in the early phase, while candidate EMPs are more likely to be derived from the positive field at the latter phases of the search as Ps_rate increases. In other words, iEFO will efficiently explore the search space during the early stages and will favor exploitation around the better solutions in the latter phases.

Finally, the iEFO employs a modified boundary check and a randomization mechanism. In the modified boundary check procedure, the electromagnets that are generated outside the boundaries are set back to the boundary values. In addition to the modification on boundary check procedure, in the random search step of the iEFO, the randomly selected EMP is regenerated within limits instead of a sequence-based approach.

As a result of the descriptions given above, Figure 2 shows the flow chart of the iEFO and illustrates the main steps of the algorithms, i.e., initialization, classification, candidate EMP generation, randomization procedure or random search, selection, and re-sorting, and the adaptive control mechanism of Ps_rate and R_rate .

Before comparing iEFO against various algorithms, the distribution of EMPs during the search process at various stages of EFO and iEFO on the sphere function ($f(\vec{X}) =$

$\sum_{i=1}^D x_i^2$) is analyzed and shown in Figure 3. It should be noted that here, the parameter settings of EFO are set as described in [14], whereas preliminary parameter settings are used for iEFO. From Figure 3, it can be concluded that the convergence performance of iEFO is better than the EFO. Specifically, the guidance of RW_j and the adaptive mechanism of iEFO improve the convergence behavior of the population.

4. Computational Results

In this section, the performance of the iEFO is analyzed and tested against various well-known metaheuristic algorithms using three sets of experiments. First, iEFO is compared with the original EFO and other recent physics-inspired algorithms, such as gravitational search algorithm (GSA) [29], electromagnetism-like algorithm (EMA) [40], central force optimization algorithm (CFO) [34], and weighted superposition attraction algorithm (WSA) [13] on different benchmark problems with various dimensions. Second, iEFO is tested against ABC, PSO, and DE using the same problem set. In order to make fair comparisons, all algorithms used in the first two sets of experiments are coded and executed in the same environment. All algorithms conduct 30 independent runs on each benchmark function, and the maximum number of function evaluation (MaxFE) is employed as the termination criterion, which is set to 320,000 for all simulations. Further, all algorithms have been simulated in the MATLAB environment and executed on the same computer with Intel Xeon CPU (2.67 GHz) and 16 GB of memory. In the last part of the experiments, the performance of the iEFO is tested against powerful variants of PSO and DE. For this part, the competitor algorithms are fully informed particle swarm (FIPS) [63], self-organizing hierarchical PSO with time-varying acceleration coefficients (HPSO-TVAC) [64], comprehensive learning PSO (CLPSO) [57], self-adapting DE (JDE) [52], adaptive DE with optional external archive (JADE) [65], and self-adaptive DE (SaDE) [62]. In this subsection, the reported results from the corresponding studies are used for comparisons.

4.1. Benchmark Functions and Parameter Settings. In order to evaluate the performance of iEFO, 13 commonly used benchmark functions [1, 5, 6, 13, 47, 48, 61, 66–72] with various dimensions, which are listed in Table 3, are used. In Table 3, D refers to the dimension of the problem. To be specific, F1 and F8 are unimodal and nonseparable functions (UN), F2, F3, F5, F6, F10, and F13 are multimodal and separable functions (MS), F4, F9, and F12 are multimodal and nonseparable (MN), and F7 and F11 are unimodal and separable functions (US). Unimodal functions have one local minimum as the global optimum, whereas multimodal functions have more than one local optimum and the number of their local optimums increases with the problem dimension exponentially. On the other hand, separable functions can be written as sum of n functions with one variable, while nonseparable functions cannot be

FIGURE 2: Flow chart of the proposed iEFO.

reformulated as the sum of subfunctions [47]. Further, F7–F10 are shifted functions and o , a shifted vector, is generated randomly in the corresponding search range. The

name, formulation, type, search space (range), and the global optimum objective function values ($f(x^*)$) are given in Table 3.

FIGURE 3: Continued.

FIGURE 3: The convergence behavior of EFO and iEFO on Sphere function. (a) Population distribution at iteration = 1 for EFO. (b) Population distribution at iteration = 1 for iEFO. (c) Population distribution at iteration = 10 for EFO. (d) Population distribution at iteration = 10 for iEFO. (e) Population distribution at iteration = 20 for EFO. (f) Population distribution at iteration = 20 for iEFO. (g) Population distribution at iteration = 30 for EFO. (h) Population distribution at iteration = 30 for iEFO.

It is worth to mention that parameter settings play a vital role in the performance of a metaheuristic algorithm [73]. The parameters of the adaptive mechanism are set to the lower and upper limits given in [REF] as $P_{S_{Min}} = 0.1$, $P_{S_{Max}} = 0.4$, $R_{Max} = 0.4$, and $R_{Min} = 0.1$. Further, preliminary tests are carried out to determine the positive and negative field ratios within the range of $P_{Field} = [0.05, 0.10]$ and $N_{Field} = [0.45, 0.50]$. According to the preliminary tests, P_{Field} and N_{Field} are set to 0.05 and 0.45, respectively. Additionally, the population size (N_{emp}) is set to 80. Table 4 presents the control parameters of all competitor algorithms. In Table 4, PopSize refers to population size of each algorithm. It should be noted that the parameter settings of the competitor algorithms are set as the original papers or published codes.

4.2. Comparison with Physics-Inspired Algorithms. To demonstrate the effectiveness of iEFO, in this subsection, the iEFO is compared with physics-inspired algorithms as EFO, GSA, EMA, CFO, and WSA. Tables 5–7 present the experimental results on benchmark problems with $D = 50$, $D = 100$, and $D = 200$. In Tables 5–7, results are given in terms of mean and standard deviation (StdDev) achieved from 30 independent runs. For a precise and pairwise comparison, the statistical significance of the differences between iEFO and competing test algorithms are determined using t -tests with a significance level of 0.05. In Tables 5–7, “+” shows that iEFO is significantly better than the compared algorithm, “ $\approx$ ” indicates that there is no significant difference between the algorithms. Last, “–” depicts that the competitor algorithms achieve better results at a level of 0.05 significance. Additionally, the overall results are given in the last rows of Tables 5–7.

As can be seen from Table 5, on the problems with $D = 50$, iEFO is better than the test algorithms on the

majority of the benchmarks. To be more specific, iEFO performs significantly better than EFO, GSA, EMA, CFO, and WSA on 6, 8, 10, 9, and 9 out of 13 functions, respectively. On F7 and F11, all algorithms obtained statistically similar results. iEFO and EFO perform similar performance on F3, F4, F5, F7, F10, F11, and F13, whereas there is no significant difference between iEFO and GSA on F7, F8, F10, F11, and F12.

On the problems with $D = 100$ (Table 6), only WSA can outperform iEFO on F2 with $D = 100$, while iEFO achieves significantly better results than WSA on benchmarks F1, F3–F6, F8, F9, F11, and F12. iEFO outperforms EFO, EMA, CFO, and WSA on 8, 10, 12, and 11 out of 13 test problems. Further, as tabulated in Table 7, for larger problems where $D = 200$, the difference between the performance of iEFO and the competitor algorithms is more significant. From Table 7, iEFO is superior to EFO, GSA, and CFO on 11, 10, 13, 12, and 13 out of 13 problems, while iEFO outperforms EMA and WSA on all test instances. This also indicates the effectiveness of iEFO on large problem instances among well-known physics-inspired metaheuristic algorithms. On the other hand, iEFO yields smaller deviation on the majority of the problem with $D = 50$, $D = 100$, and $D = 200$. This also shows that iEFO surpasses all the competitor algorithms in terms of robustness.

Moreover, the convergence performance of iEFO and competitor algorithms are given in Figure 4 for selected benchmark problems. In Figure 4, the population means are plotted against function evaluation number. It can be concluded from Figure 4 that iEFO indicates better convergence performance than the test algorithms. The results also reveal that the proposed framework appears to be highly competitive in terms of convergence speed.

Lastly, the CPU time results of all algorithms are tabulated in Table 8. Table 8 summarizes the average of

TABLE 3: Benchmark functions used for the comparisons.

Label	Name	Formulation	Type	Range	$f(x^*)$
F1	Rosenbrock	$f_1(\vec{X}) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	UN	$[-2.048, 2.048]^D$	0
F2	Ackley	$f_2(\vec{X}) = -20 \exp(-0.2 \sqrt{(1/D) \sum_{i=1}^D x_i^2}) - \exp((1/D) \sum_{i=1}^D \cos(2\pi x_i)) + 20 + e$	MS	$[-32.768, 32.768]^D$	0
F3	Rastrigin	$f_3(\vec{X}) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$	MS	$[-5.12, 5.12]^D$	0
F4	Griewank	$f_4(\vec{X}) = (1/4000) \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos(x_i / \sqrt{i}) + 1$	MN	$[-600, 600]^D$	0
F5	Weierstrass	$f_5(\vec{X}) = \sum_{i=1}^D (\sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k (x_i + 0.5))]) - D \sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k 0.5)]$, $a = 0.5$, $b = 3$, $k_{\max} = 20$	MS	$[-0.5, 0.5]^D$	0
F6	Schwefel 2.26	$f_6(\vec{X}) = \sum_{i=1}^D -x_i \sin(\sqrt{ x_i })$	MS	$[-500, 500]^D$	$-418.98 \times D$
F7	Shifted Sphere	$f_7(\vec{X}) = \sum_{i=1}^D z_i^2 - f_{\text{bias}}$, $z = x - o$, $f_{\text{bias}} = -450$	US	$[-100, 100]^D$	f_{bias}
F8	Shifted Schwefel 1.2	$f_8(\vec{X}) = \sum_{i=1}^D (\sum_{j=1}^i z_j) + f_{\text{bias}}$, $z = x - o$, $f_{\text{bias}} = -450$	UN	$[-100, 100]^D$	f_{bias}
F9	Shifted Rosenbrock	$f_9(\vec{X}) = \sum_{i=1}^{D-1} (100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2) + f_{\text{bias}}$, $z = x - o + 1$, $f_{\text{bias}} = 390$	MN	$[-100, 100]^D$	f_{bias}
F10	Shifted Rastrigin	$f_{10}(\vec{X}) = \sum_{i=1}^D [z_i^2 - 10 \cos(2\pi z_i) + 10] + f_{\text{bias}}$, $z = x - o$, $f_{\text{bias}} = -330$	MS	$[-5, 5]^D$	f_{bias}
F11	Step	$f_{11}(\vec{X}) = \sum_{i=1}^D (x_i + 0.5)^2$	US	$[-100, 100]^D$	0
F12	Penalized 2	$f_{12}(\vec{X}) = (1/10) \{ \sin^2(3\pi x_1) + \sum_{i=1}^{D-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + (x_n - 1)^2 [1 + \sin^2(2\pi x_{i+1})] \} + \sum_{i=1}^D u(x_i, 5, 100, 4)$	MN	$[-50, 50]^D$	0
F13	Alpine	$f_{13}(\vec{X}) = \sum_{i=1}^D x_i \cdot \sin(x_i) + 0.1 \cdot x_i $	MS	$[-10, 10]^D$	0

TABLE 4: Parameter settings of algorithms.

Algorithm	Parameter setting
EFO [14]	$N_emp = 50$, $P_field = 0.1$, $N_field = 0.45$, $Ps_rate = 0.2$, $R_rate = 0.4$,
GSA [29]	$PopSize = 50$, $G_0 = 100$, $\alpha = 20$, $K_0 = PopSize$
EMA [40]	$PopSize = 50$, $\delta = 0.01$
CFO [34]	$PopSize = 50$, $G = 2$, $\alpha = 2$, $\beta = 2$
WSA [13]	$PopSize = 20$, $\tau = 0.8$, $sl_o = 0.035$, $\varphi = 0.001$, $\lambda = 0.75$
ABC [74]	$PopSize = 40$, $Limit = PopSize \times D$
PSO [39]	$PopSize = 40$, $w = 1.193$, $c1 = 1.193$, $c2 = 0.721$
DE [75]	$PopSize = 40$, $F = 0.9$, $CR = 0.9$

CPU times on different dimensions in terms of mean and standard deviation (StdDev). As can be seen from Table 8, the CPU times of iEFO, EFO, GSA, EMA, CFO, and WSA are very close to each other. In other words, there is no significance difference between iEFO and other test algorithms.

Furthermore, the average rankings based on non-parametric Friedman's test of all compared algorithms are given in Table 9. As shown in Table 9, iEFO ranks the first on all dimensions. The test statistics and the p -values also indicate the significant difference between iEFO and other algorithms. Overall, the experimental results in this subsection indicate the outperforming performance of iEFO against other physics-inspired algorithms.

4.3. Comparison with ABC, PSO, and DE. In this subsection, the performance comparisons on the solution quality are conducted among iEFO and ABC, PSO, and DE. The results shown in Tables 10–12 in terms of mean and StdDev of the solutions obtained from 30 independent runs. Similar to the previous subsection, t -tests are carried out to test the significance of the results. As described in Table 10, for the benchmark functions with $D = 50$, it is worth noting that iEFO is significantly superior to all competitor algorithms on most of the cases. iEFO and ABC show similar performance on F1, F6, F7, F10, and F11, and iEFO performs significantly better than ABC on all other benchmarks. Further, iEFO outperforms PSO and DE on 8 and 6 out of 13 problems, respectively. For the functions with $D = 100$, according to the overall performance given in the last row of Table 11, it is clear that iEFO wins in 9, 10, and 10 problems against ABC, PSO, and DE, respectively. iEFO and ABC tie on four functions (F1, F7, F9, and F13), iEFO and PSO tie on three functions (F5, F7, and F10), and similarly there is no significance difference between iEFO and DE on F1, F7, and F10. As described in Table 12, where $D = 200$, iEFO superior to ABC, PSO, and DE on 11, 12, and 12 out of 13 functions, respectively. Further, iEFO generally reaches smaller deviations when compared to the test algorithms, which validates the robustness of the proposed iEFO. To intuitively verify the significance difference between iEFO and other three state-of-the-art algorithms, Friedman's tests are carried out similar to the previous subsection.

Table 13 shows Friedman's test rankings test of all compared algorithms. From Table 13, it is clear that iEFO is the best algorithm among ABC, PSO, and DE. Further, the performance of iEFO is even better as the problem size increases.

Summarizing the above statements, iEFO achieves high-quality results in terms of solution quality and robustness when compared to state-of-the-art algorithms, i.e., ABC, PSO, and DE. In other words, the exploitation and exploration of iEFO are well balanced.

4.4. Comparison with PSO and DE Variants. This subsection covers the computational results of iEFO against state-of-the-art PSO and DE variants. Since the reported results are taken directly from the corresponding papers, statistical significance tests between algorithms are not carried out. Tables 14 and 15 present computational results. The results for the test cases with $D = 30$ are listed in Tables 13 and 14. Some of the test problems are not covered in the comparisons, as corresponding results were not given in the original studies. In Tables 14 and 15, the best mean values are indicated in bold. As can be seen from Table 14, iEFO outperforms FIPS, CLPSO, and HPSO-HVAC on F1–F3, F6, and F13. Further, FIPS is better than iEFO on F4 and HPSO-HVAC is superior to iEFO on F12. All algorithms reach the global optimum for all runs on F11. From Table 15, it is clear that iEFO outperforms JDE and SaDE on all benchmark problems. On the other hand, JADE achieves the best results on F2, F11, and F12. These results also reveal the effectiveness of iEFO when compared to novel PSO and DE variants.

5. Conclusion

In this paper, an improved version of the EFO (iEFO) is introduced to solve the global optimization problem. The iEFO enriches the solution search strategy of the original EFO by improving two main steps of the algorithm. First, roulette wheel selection strategy embedded new solution generation function is used to provide more chance to use of fittest electromagnetic particles in the candidate solution generation step. Second, an adaptive control mechanism for the algorithmic parameters is used to increase the selection probability of better solution and reduce the selection probability of worse solution throughout the search process. Besides both two improvements, the boundary control and randomization procedure is modified. To test the performance of the proposed algorithm, a well-known benchmark problem set is used in computational studies. To show the efficiency of the iEFO, three sets of comparisons are carried out. First, the iEFO is compared with original EFO and existing physics-inspired metaheuristic algorithms. Second, three well-known metaheuristic algorithms (ABC, DE, and PSO), whose performances are demonstrated on global optimization problem in many researches, are taken into account. Finally, the iEFO is compared with the improved variants of the DE and PSO. As a result of

TABLE 5: Comparisons of iEFO with physics-inspired algorithms on benchmark problems with $D = 50$.

Problem	iEFO			EFO			GSA			EMA			CFO			WSA		
	Mean	StdDev	t -test	Mean	StdDev	t -test	Mean	StdDev	t -test	Mean	StdDev	t -test	Mean	StdDev	t -test	Mean	StdDev	t -test
F1	2.64E+01	3.12E-01	5.29E+02	3.20E+02	3.20E+02	+	4.90E+01	8.29E+00	+	5.33E+04	2.03E+03	+	8.23E+03	1.23E+02	+	9.23E+03	3.55E+02	+
F2	4.96E-14	8.32E-15	9.07E-13	4.91E-13	4.91E-13	+	7.44E-13	3.90E-14	+	2.93E-09	7.19E-12	+	7.33E-12	5.29E-14	+	8.02E-12	4.92E-13	+
F3	4.92E-16	6.93E-16	4.33E-14	6.02E-13	6.02E-13	$\approx$	7.03E-14	9.91E-14	+	8.18E-10	4.94E-11	+	7.17E-12	9.62E+12	$\approx$	6.30E-13	5.93E-14	+
F4	1.59E-16	5.63E-17	8.93E-15	2.77E-11	2.77E-11	$\approx$	3.29E-15	6.94E-15	+	5.03E-11	4.32E-11	+	3.09E-11	2.59E-14	+	1.92E-11	7.04E-11	$\approx$
F5	6.31E-16	3.08E-17	1.09E-15	8.03E-15	8.03E-15	$\approx$	6.93E-15	5.49E-16	+	6.81E-12	8.61E-12	+	7.81E-14	3.99E-16	+	8.32E-16	4.29E-17	+
F6	-2.09E+04	2.41E+00	-1.99E+04	4.30E+01	4.30E+01	+	-2.04E+04	3.41E+00	+	-1.99E+04	1.12E+01	+	-2.01E+04	4.31E+02	+	-1.76E+04	2.23E+03	+
F7	-4.50E+02	0.00E+00	-4.50E+02	0.00E+00	0.00E+00	$\approx$	-4.50E+02	0.00E+00	$\approx$	-4.50E+02	0.00E+00	$\approx$	-4.50E+02	0.00E+00	$\approx$	-4.50E+02	0.00E+00	$\approx$
F8	2.94E+04	1.59E+03	5.30E+04	4.49E+02	4.49E+02	+	3.04E+04	8.02E+03	+	8.44E+06	5.14E+03	+	6.11E+05	8.92E+03	+	4.92E+05	8.85E+04	+
F9	3.15E+02	2.31E+00	9.51E+03	2.41E+01	2.41E+01	+	3.28E+02	1.02E+01	+	3.90E+04	9.54E+02	+	7.32E+04	2.56E+02	+	3.20E+04	4.91E+02	+
F10	-3.30E+02	0.00E+00	-3.30E+02	1.00E+01	4.32E+00	$\approx$	-3.30E+02	4.32E+00	$\approx$	-2.88E+02	4.30E+01	+	-3.30E+02	1.20E+01	$\approx$	-3.30E+02	2.36E+02	$\approx$
F11	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	$\approx$	0.00E+00	0.00E+00	$\approx$	0.00E+00	0.00E+00	$\approx$	0.00E+00	0.00E+00	$\approx$	0.00E+00	0.00E+00	$\approx$
F12	5.81E-18	8.13E-17	8.84E-17	1.95E-16	6.01E-17	+	1.53E-17	6.01E-17	+	4.33E-13	6.02E-08	$\approx$	5.91E-15	9.27E-19	+	7.20E-17	9.03E-19	+
F13	2.06E-24	1.22E-25	8.52E-23	2.91E-22	2.91E-22	$\approx$	3.77E-22	6.58E-23	+	8.58E-15	8.36E-15	+	7.84E-21	3.03E-21	+	4.04E-21	5.39E-22	+
+/- / $\approx$				6/0/7	8/0/5					10/0/3			9/0/4				9/0/4	

TABLE 6: Comparisons of iEFO with physics-inspired algorithms on benchmark problems with $D = 100$.

Problem	iEFO			EFO			GSA			EMA			CFO			WSA		
	Mean	StdDev	t-test	Mean	StdDev	t-test	Mean	StdDev	t-test	Mean	StdDev	t-test	Mean	StdDev	t-test	Mean	StdDev	t-test
F1	9.83E+01	6.39E+00	7.94E+02	5.72E+01	5.72E+01	+	1.34E+02	2.11E+01	+	6.88E+05	3.86E+02	+	1.14E+04	6.94E+02	+	2.24E+05	5.06E+02	+
F2	4.03E-11	1.28E-12	8.11E-12	9.48E-11	9.48E-11	≈	6.23E-12	4.18E-14	-	8.21E-08	8.31E-09	+	8.21E-11	2.60E-11	+	6.73E-11	4.77E-13	+
F3	7.31E-14	8.23E-15	9.34E-11	9.34E-12	9.34E-12	+	6.26E-13	2.10E-13	+	5.54E-09	9.18E-10	+	8.51E-11	4.99E-12	+	3.53E-11	2.28E-12	+
F4	6.04E-14	4.91E-15	4.13E-11	6.49E-13	6.49E-13	+	2.95E-13	7.98E-14	+	7.59E-10	3.13E-10	+	2.11E-11	6.82E-12	+	4.18E-11	1.90E-10	≈
F5	2.04E-13	7.17E-13	3.90E-13	1.44E-11	1.44E-11	≈	1.01E-12	6.80E-13	+	4.63E-10	9.40E-10	≈	7.43E-12	4.47E-13	+	9.75E-13	4.17E-16	+
F6	-1.12E+04	3.55E+00	-8.40E+03	3.02E+01	7.35E+00	+	-9.34E+03	7.35E+00	+	-5.68E+01	6.00E+01	+	-6.31E+03	3.85E+00	+	-8.21E+02	2.07E+01	+
F7	-4.50E+02	8.34E-02	-3.91E+02	4.23E+00	4.23E+00	+	-4.50E+02	2.30E+01	≈	-6.20E+01	7.60E+00	+	-6.74E+01	5.46E+00	+	-5.93E+01	5.43E+00	+
F8	8.92E+04	8.34E+04	9.34E+04	7.33E+05	1.86E+01	≈	4.73E+05	1.86E+01	+	9.95E+04	5.10E+01	+	9.98E+06	8.35E+02	+	5.36E+07	5.21E+01	+
F9	9.44E+02	6.93E+01	2.90E+04	8.93E+02	9.36E+01	+	3.11E+03	9.36E+01	+	4.09E+05	1.64E+03	+	3.50E+06	8.56E+01	+	3.82E+06	4.24E+02	+
F10	-3.30E+02	5.92E-06	-3.30E+02	5.29E-01	4.14E-03	≈	-3.30E+02	4.14E-03	≈	-1.66E+01	3.92E-02	+	-9.01E+01	6.91E+00	+	-2.35E+00	1.59E+00	+
F11	1.35E+01	7.30E+01	5.12E+02	3.20E+02	7.49E+00	+	7.13E+01	7.49E+00	+	4.02E+02	3.96E+00	+	2.34E+02	3.31E+00	+	9.83E+03	7.29E+00	+
F12	4.18E-13	1.05E-13	9.12E-12	5.02E-12	4.02E-14	+	5.25E-13	4.02E-14	+	8.41E-11	8.40E-12	+	1.15E-12	5.09E-13	+	3.12E-12	7.37E-12	≈
F13	7.07E-20	1.66E-22	7.18E-20	5.27E-15	9.02E-19	≈	7.93E-20	9.02E-19	≈	2.12E-15	6.84E-14	≈	3.74E-16	6.60E-15	≈	2.95E-16	1.51E-17	+
+/- / ≈				8/0/5	9/1/3			10/0/3			12/0/1			11/0/2				

TABLE 7: Comparisons of iEFO with physics-inspired algorithms on benchmark problems with $D = 200$.

Problem	iEFO			EFO			GSA			EMA			CFO			WSA		
	Mean	StdDev		Mean	StdDev	t -test	Mean	StdDev		Mean	StdDev	t -test	Mean	StdDev		Mean	StdDev	t -test
F1	1.17E+02	6.26E+01		3.82E+03	6.86E+01	+	8.95E+04	5.30E+03	+	4.53E+07	3.02E+02	+	4.63E+05	3.96E+02	+	1.41E+07	4.04E+02	+
F2	3.91E-05	8.22E-07		6.50E-04	8.62E-06	+	3.83E-05	3.06E-05	+	6.01E-04	9.02E-06	+	2.68E-04	2.28E-04	+	4.96E-05	3.81E-07	+
F3	1.32E-04	3.91E-06		5.96E-03	7.74E-07	+	8.89E-03	1.37E-04	+	1.24E-03	9.60E-04	+	2.29E-01	4.66E-03	+	8.69E-02	7.94E-05	+
F4	9.81E-10	4.81E-09		6.80E-09	4.50E-08	$\approx$	3.57E-09	6.54E-10	+	3.04E-08	3.92E-10	+	2.05E-08	3.85E-07	$\approx$	2.82E-09	6.58E-11	+
F5	1.85E-04	7.49E-05		5.86E-03	3.53E-04	+	6.26E-03	3.11E-04	+	4.21E-03	9.94E-04	+	5.59E-03	7.17E-04	+	4.96E-03	3.74E-05	+
F6	-8.12E+04	3.14E+01		-5.92E+03	5.57E+02	+	-7.94E+02	2.84E+01	+	-3.86E+03	8.43E+01	+	-6.27E+03	6.49E+01	+	-2.33E+04	4.53E+00	+
F7	3.12E+05	4.73E+02		1.92E+06	5.60E+03	+	9.37E+05	4.36E+02	+	5.02E+08	6.75E+02	+	5.07E+05	7.49E+02	+	2.87E+06	7.99E+02	+
F8	3.44E+05	7.22E+03		9.15E+07	6.08E+03	+	2.80E+06	9.36E+03	+	7.06E+06	1.25E+03	+	8.33E+08	3.84E+02	+	1.48E+09	5.68E+04	+
F9	4.38E+06	8.04E+04		9.92E+06	7.80E+02	+	9.48E+06	9.20E+02	+	6.28E+06	9.62E+03	+	1.43E+08	7.55E+03	+	1.33E+08	4.31E+03	+
F10	4.33E+02	3.34E+01		5.24E+02	1.24E+01	+	7.70E+02	8.65E+02	$\approx$	7.16E+02	7.21E+00	+	6.81E+02	5.84E+01	+	1.80E+03	7.30E+01	+
F11	3.28E+02	8.30E+01		4.64E+03	8.51E+01	+	7.10E+02	6.19E+01	+	2.54E+04	5.28E+03	+	1.56E+04	4.93E+02	+	6.95E+04	1.35E+02	+
F12	1.82E-11	7.85E-11		1.05E-09	9.20E-10	+	1.50E-08	8.48E-10	+	2.07E-09	4.29E-11	+	3.26E-08	7.44E-13	+	1.02E-10	4.43E-11	+
F13	2.03E-18	5.73E-15		7.37E-16	1.66E-14	$\approx$	2.32E-17	5.56E-14	$\approx$	1.37E-13	9.61E-14	+	8.03E-15	6.90E-15	+	2.16E-14	8.92E-16	+
+/-/ $\approx$								10/0/3			13/0/0			12/0/1			13/0/0

(a)

(b)

(c)

(d)

FIGURE 4: Continued.

FIGURE 4: Convergence performance of iEFO and competitor algorithms on $D = 100$.

TABLE 8: CPU time comparison of iEFO and competitor algorithms.

Algorithm	$D = 50$		$D = 100$		$D = 200$	
	Mean	StdDev	Mean	StdDev	Mean	StdDev
iEFO	13.83	1.43	16.73	1.14	18.54	1.92
EFO	13.11	2.08	15.84	1.56	17.21	1.77
GSA	12.91	1.74	15.18	1.82	18.37	1.98
EMA	13.50	1.68	15.29	1.70	17.84	2.04
CFO	14.79	1.97	16.75	1.94	18.92	1.96
WSA	12.46	1.56	15.22	1.81	17.33	1.89

TABLE 9: Friedman-test results for the iEFO and physics-inspired algorithms.

Rank	Average		Dimension ($D = 50$)		Dimension ($D = 100$)		Dimension ($D = 200$)	
	Algorithm	Ranking	Algorithm	Ranking	Algorithm	Ranking	Algorithm	Ranking
1	iEFO	1.29	iEFO	1.54	iEFO	1.27	iEFO	1.08
2	GSA	2.78	GSA	2.62	GSA	2.27	GSA	3.46
3	EFO	3.29	EFO	3.12	EFO	3.15	EFO	3.62
4	CFO	4.28	WSA	3.92	CFO	4.15	WSA	4.23
5	WSA	4.33	CFO	4.38	WSA	4.85	EMA	4.31
6	EMA	5.01	EMA	5.42	EMA	5.31	CFO	4.31
		Statistic	35.05		45.01		35.05	
		p value	<0.001		<0.001		<0.001	

TABLE 10: Comparisons of iEFO with ABC, PSO, and DE on benchmark problems with $D = 50$.

Problem	iEFO		ABC		t -test	PSO		t -test	DE		t -test
	Mean	StdDev	Mean	StdDev		Mean	StdDev		Mean	StdDev	
F1	$2.64E+01$	$3.12E-01$	$3.02E+01$	$1.11E+01$	$\approx$	$3.54E+01$	$1.44E+01$	+	$3.32E+01$	$1.05E+01$	+
F2	$4.96E-14$	$8.32E-15$	$1.21E-13$	$1.51E-14$	+	$7.90E-14$	$1.89E-14$	+	$8.06E-14$	$8.95E-15$	+
F3	$4.92E-16$	$6.93E-16$	$2.31E-11$	$7.29E-12$	+	$8.44E-14$	$6.86E-12$	$\approx$	$9.11E-14$	$1.95E-12$	$\approx$
F4	$1.59E-16$	$5.63E-17$	$4.88E-12$	$2.85E-13$	+	$1.04E-13$	$2.11E-13$	+	$5.89E-16$	$1.55E-13$	$\approx$

TABLE 10: Continued.

Problem	iEFO		ABC		t -test	PSO		t -test	DE		t -test
	Mean	StdDev	Mean	StdDev		Mean	StdDev		Mean	StdDev	
F5	$6.31E-16$	$3.08E-17$	$3.66E-14$	$2.01E-14$	+	$7.70E-15$	$1.23E-16$	+	$8.37E-15$	$8.59E-15$	+
F6	$-2.09E+04$	$2.41E+00$	$-2.09E+04$	$6.01E+00$	$\approx$	$-2.09E+04$	$6.01E+00$	$\approx$	$-2.09E+04$	$4.15E+00$	$\approx$
F7	$-4.50E+02$	$0.00E+00$	$-4.50E+02$	$6.90E-14$	$\approx$	$-4.50E+02$	$7.11E-14$	$\approx$	$-4.50E+02$	$8.27E-15$	$\approx$
F8	$2.94E+04$	$1.59E+03$	$3.29E+04$	$1.43E+03$	+	$3.10E+04$	$1.11E+03$	+	$3.89E+04$	$1.44E+03$	+
F9	$3.15E+02$	$2.31E+00$	$5.88E+02$	$4.52E+00$	+	$4.19E+02$	$5.01E+00$	+	$3.87E+02$	$2.90E+00$	+
F10	$-3.30E+02$	$0.00E+00$	$-3.30E+02$	$3.02E-14$	$\approx$	$-3.30E+02$	$2.86E-14$	$\approx$	$-3.30E+02$	$7.40E-15$	$\approx$
F11	$0.00E+00$	$0.00E+00$	$0.00E+00$	$0.00E+00$	$\approx$	$0.00E+00$	$0.00E+00$	$\approx$	$0.00E+00$	$0.00E+00$	$\approx$
F12	$5.81E-18$	$8.13E-17$	$5.44E-17$	$9.44E-17$	+	$9.05E-17$	$5.68E-17$	+	$1.59E-17$	$6.27E-16$	$\approx$
F13	$2.06E-24$	$1.22E-25$	$3.25E-23$	$2.65E-23$	+	$6.13E-24$	$8.66E-24$	+	$7.95E-24$	$9.06E-24$	+
+/-/ $\approx$				8/0/5			8/0/5			6/0/7	

TABLE 11: Comparisons of iEFO with ABC, PSO, and DE on benchmark problems with $D = 100$.

Problem	iEFO		ABC		t -test	PSO		t -test	DE		t -test
	Mean	StdDev	Mean	StdDev		Mean	StdDev		Mean	StdDev	
F1	$9.83E+01$	$6.39E+00$	$1.44E+02$	$4.71E+02$	$\approx$	$1.15E+03$	$9.00E+02$	+	$1.10E+02$	$2.31E+03$	$\approx$
F2	$4.03E-12$	$1.28E-12$	$6.34E-07$	$7.44E-08$	+	$3.04E-09$	$8.43E-11$	+	$6.41E-11$	$4.02E-12$	+
F3	$7.31E-14$	$8.23E-15$	$2.43E-05$	$1.00E-06$	+	$5.84E-09$	$3.61E-10$	+	$1.21E-12$	$7.26E-13$	+
F4	$6.04E-14$	$4.91E-15$	$1.11E-04$	$7.33E-05$	+	$4.22E-09$	$2.68E-10$	+	$3.54E-12$	$7.22E-14$	+
F5	$2.04E-13$	$7.17E-13$	$1.92E-04$	$8.90E-15$	+	$3.96E-13$	$4.92E-14$	$\approx$	$5.40E-11$	$3.44E-14$	+
F6	$-1.12E+04$	$3.55E+00$	$-4.02E+03$	$2.11E+02$	+	$-4.11E+03$	$1.53E+02$	+	$-4.19E+03$	$6.24E+01$	+
F7	$-4.50E+02$	$8.34E-02$	$-4.50E+02$	$5.32E+01$	$\approx$	$-4.50E+02$	$8.53E+00$	$\approx$	$-4.50E+02$	$4.90E+00$	$\approx$
F8	$8.92E+04$	$8.34E+04$	$2.35E+05$	$3.05E+04$	+	$1.65E+05$	$1.05E+04$	+	$1.66E+05$	$7.77E+03$	+
F9	$9.44E+02$	$6.93E+01$	$9.01E+02$	$2.11E+02$	$\approx$	$4.01E+03$	$2.98E+01$	+	$4.52E+03$	$9.01E+01$	+
F10	$-3.30E+02$	$5.92E-06$	$-3.22E+02$	$3.68E-05$	+	$-3.30E+02$	$7.20E-07$	$\approx$	$-3.30E+02$	$6.82E-14$	$\approx$
F11	$1.35E+01$	$7.30E+01$	$5.37E+01$	$8.73E+00$	+	$6.51E+01$	$6.89E+00$	+	$5.43E+01$	$9.52E+00$	+
F12	$4.18E-13$	$1.05E-13$	$6.84E-11$	$5.32E-13$	+	$7.16E-13$	$3.18E-14$	+	$1.15E-12$	$3.91E-13$	+
F13	$7.07E-21$	$1.66E-22$	$9.05E-19$	$3.04E-18$	$\approx$	$9.04E-20$	$1.22E-21$	+	$7.89E-19$	$1.06E-19$	+
+/-/ $\approx$				9/0/4			10/0/3			10/0/3	

TABLE 12: Comparisons of iEFO with ABC, PSO, and DE on benchmark problems with $D = 200$.

Problem	iEFO		ABC		t -test	PSO		t -test	DE		t -test
	Mean	StdDev	Mean	StdDev		Mean	StdDev		Mean	StdDev	
F1	$1.17E+02$	$6.26E+01$	$4.32E+02$	$1.04E+02$	+	$4.47E+02$	$6.20E+01$	+	$4.58E+02$	$3.94E+01$	+
F2	$3.91E-05$	$8.22E-07$	$4.44E-02$	$5.28E-03$	+	$6.68E-05$	$8.54E-06$	+	$1.47E-04$	$6.23E-05$	+
F3	$1.32E-04$	$3.91E-03$	$3.17E+01$	$6.03E+00$	+	$6.51E+00$	$2.44E+00$	+	$3.91E-01$	$6.16E-02$	+
F4	$9.81E-10$	$4.81E-09$	$5.25E-04$	$8.30E-06$	+	$1.79E-08$	$7.19E-09$	+	$6.05E-08$	$3.65E-09$	+
F5	$1.85E-04$	$7.49E-05$	$1.51E-01$	$3.55E-03$	+	$9.76E-03$	$1.13E-04$	+	$1.83E-02$	$7.04E-03$	+
F6	$-8.12E+04$	$3.14E+01$	$-7.59E+04$	$5.01E+02$	+	$-8.15E+04$	$4.99E+03$	$\approx$	$-7.37E+04$	$8.34E+02$	+
F7	$6.12E+05$	$4.73E+02$	$8.10E+05$	$1.46E+04$	+	$8.73E+05$	$9.49E+04$	+	$8.62E+05$	$2.09E+04$	+
F8	$3.44E+05$	$7.22E+03$	$2.55E+06$	$5.29E+04$	+	$2.93E+06$	$5.18E+05$	+	$2.98E+06$	$7.35E+05$	+
F9	$4.38E+10$	$8.04E+04$	$8.07E+11$	$5.04E+07$	+	$7.87E+11$	$4.91E+06$	+	$7.61E+11$	$4.33E+05$	+
F10	$4.33E+02$	$3.34E+01$	$1.57E+02$	$2.49E+03$	$\approx$	$3.76E+03$	$7.37E+02$	+	$3.76E+03$	$2.54E+02$	+
F11	$3.28E+02$	$8.30E+01$	$8.15E+04$	$9.90E+02$	+	$5.39E+03$	$3.04E+02$	+	$1.04E+03$	$9.46E+02$	+
F12	$1.82E-11$	$7.85E-10$	$2.01E-09$	$5.33E-12$	+	$4.88E-11$	$9.92E-12$	$\approx$	$6.22E-11$	$9.04E-13$	$\approx$
F13	$2.03E-18$	$5.73E-18$	$5.82E-16$	$4.40E-15$	$\approx$	$4.72E-16$	$2.81E-17$	+	$1.92E-17$	$4.23E-18$	+
+/-/ $\approx$				11/0/2			11/0/2			12/0/1	

TABLE 13: Friedman-test results for the iEFO and ABC, PSO, and DE.

Rank	Average		Dimension ($D = 50$)		Dimension ($D = 100$)		Dimension ($D = 200$)	
	Algorithm	Ranking	Algorithm	Ranking	Algorithm	Ranking	Algorithm	Ranking
1	iEFO	1.29	iEFO	1.46	iEFO	1.27	iEFO	1.15
2	PSO	2.67	PSO	2.62	DE	2.58	PSO	2.65
3	DE	2.74	DE	2.69	PSO	2.73	DE	2.96
4	ABC	3.29	ABC	3.23	ABC	3.42	ABC	3.23
		Statistic		12.97		18.92		20.15
		p value		0.009		0.001		<0.001

TABLE 14: Comparisons of iEFO with PSO variants on benchmark problems with $D = 30$.

Problem	MaxFE	iEFO		FIPS		HPSO-TVAC		CLPSO	
		Mean	StdDev	Mean	StdDev	Mean	StdDev	Mean	StdDev
F1	200,000	5.23E+00	2.34E+00	2.51E+01	5.10E-01	2.39E+01	2.65E+01	1.13E+01	9.85E+00
F2	200,000	9.19E-15	5.90E-14	2.33E-07	7.19E-08	7.29E-14	3.00E-14	3.66E-07	7.57E-08
F3	200,000	7.02E-05	3.48E-05	6.51E+01	1.33E+01	9.43E+00	3.48E+00	9.05E-05	1.25E-04
F4	200,000	9.55E-12	1.82E-13	9.01E-12	1.84E-11	9.75E-03	8.33E-03	9.02E-09	8.57E+09
F6	200,000	8.02E-05	7.61E-04	9.93E+02	5.09E+02	1.59E+03	3.26E+02	3.82E-04	1.28E-05
F11	200,000	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F12	200,000	3.90E-20	4.50E-18	2.70E-14	1.57E-14	2.79E-28	2.18E-28	1.25E-12	9.45E-12
F13	200,000	5.23E+00	2.34E+00	2.51E+01	5.10E-01	2.39E+01	2.65E+01	1.13E+01	9.85E+00

TABLE 15: Comparisons of iEFO with DE variants on benchmark problems with $D = 30$.

Problem	MaxFE	iEFO		JDE		JADE		SaDE	
		Mean	StdDev	Mean	StdDev	Mean	StdDev	Mean	StdDev
F1	300,000	2.94E-01	6.23E-01	1.30E+01	1.40E+01	3.20E-01	1.10E+00	2.10E+01	7.70E+01
F2	50,000	5.12E-09	8.24E-08	2.37E-04	7.10E-05	3.35E-09	2.84E-09	3.81E-06	8.26E-07
F3	100,000	1.63E-09	5.22E-09	2.37E-04	7.10E-05	3.35E-09	2.84E-09	3.81E-06	8.26E-07
F4	50,000	9.48E-10	7.19E-09	7.29E-06	1.05E-05	1.57E-08	1.09E-07	2.52E-09	1.24E-09
F6	100,000	7.92E-11	4.20E-11	1.70E-10	2.62E-10	2.62E-04	3.59E-04	1.13E-08	1.08E-08
F11	10,000	9.49E+00	3.11E+00	6.13E+02	1.72E+02	5.62E+00	1.87E+00	5.07E+01	1.34E+01
F12	50,000	1.99E-10	9.19E-11	1.80E-05	1.42E-05	1.87E-10	1.09E-09	1.93E-09	1.53E-09
F13	300,000	5.38E-10	3.61E-09	6.08E-10	8.36E-10	2.78E-05	8.43E-06	2.94E-06	3.47E-06

the extensive statistical analyses, it can be expressed that the proposed iEFO provides efficient solutions and superior to competitor algorithms on most of the problem instances. As a future work, the binary or integer version of the iEFO can be studied for combinatorial problems, such as scheduling problems, vehicle routing problems, knapsack problems, and set covering problems. Further, hybridizing the iEFO with other metaheuristic algorithms will be vital research direction.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The author declares that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

The author would like to thank the editor and anonymous referees for their helpful comments and suggestions in improving the presentation of this paper.

References

- [1] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp Swarm algorithm: a bio-inspired optimizer for engineering design problems," *Advances in Engineering Software*, vol. 114, pp. 163–191, 2017.
- [2] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.
- [3] G.-G. Wang, X. Zhao, and S. Deb, "A novel monarch butterfly optimization with greedy strategy and self-adaptive," in *Proceedings of the Second International Conference on Soft Computing and Machine Intelligence (ISCMI), 2015*, pp. 45–50, IEEE, Hong Kong, China, June 2015.
- [4] H. Shayanfar and F. S. Gharehchopogh, "Farmland fertility: a new metaheuristic algorithm for solving continuous optimization problems," *Applied Soft Computing*, vol. 71, pp. 728–746, 2018.
- [5] M. Jain, V. Singh, and A. Rani, "A novel nature-inspired algorithm for optimization: squirrel search algorithm," *Swarm and Evolutionary Computation*, vol. 44, pp. 148–175, 2019.
- [6] W. Zhao, L. Wang, and Z. Zhang, "Atom search optimization and its application to solve a hydrogeologic parameter estimation problem," *Knowledge-Based Systems*, vol. 163, pp. 283–304, 2019.
- [7] G. Dhiman and V. Kumar, "Emperor penguin optimizer: a bio-inspired algorithm for engineering problems," *Knowledge-Based Systems*, vol. 159, pp. 20–50, 2018.
- [8] A. Biswas, K. Mishra, S. Tiwari, and A. Misra, "Physics-inspired optimization algorithms: a survey," *Journal of Optimization*, vol. 2013, Article ID 438152, 16 pages, 2013.
- [9] U. Can and B. Alatas, "Physics based metaheuristic algorithms for global optimization," *American Journal of Information Science and Computer Engineering*, vol. 1, no. 3, pp. 94–106, 2015.
- [10] M. Tahani and N. Babayan, "Flow Regime Algorithm (FRA): a physics-based meta-heuristics algorithm," *Knowledge and Information Systems*, pp. 1–38, 2018.
- [11] A. Wedyan, J. Whalley, and A. Narayanan, "Hydrological cycle algorithm for continuous optimization problems,"

- Journal of Optimization*, vol. 2017, Article ID 3828420, 25 pages, 2017.
- [12] A. Kaveh and A. Dadras, "A novel meta-heuristic optimization algorithm: thermal exchange optimization," *Advances in Engineering Software*, vol. 110, pp. 69–84, 2017.
 - [13] A. Baykasoglu and S. Akpinar, "Weighted Superposition Attraction (WSA): a swarm intelligence algorithm for optimization problems—part 1: unconstrained optimization," *Applied Soft Computing*, vol. 56, pp. 520–540, 2017.
 - [14] H. Abedinpourshotorban, S. Mariyam Shamsuddin, Z. Beheshti, and D. N. A. Jawawi, "Electromagnetic field optimization: a physics-inspired metaheuristic optimization algorithm," *Swarm and Evolutionary Computation*, vol. 26, pp. 8–22, 2016.
 - [15] T. Biyanto, M. Syamsi, H. Fibrianto et al., "Optimization of energy efficiency and conservation in green building design using Duelist, Killer-Whale and Rain-Water Algorithms," *IOP Conference Series: Materials Science and Engineering*, vol. 267, no. 1, article 012036, 2017.
 - [16] B. Javidy, A. Hatamlou, and S. Mirjalili, "Ions motion algorithm for solving optimization problems," *Applied Soft Computing*, vol. 32, pp. 72–79, 2015.
 - [17] A. Kaveh and V. R. Mahdavi, "Colliding bodies optimization: a novel meta-heuristic method," *Computers and Structures*, vol. 139, pp. 18–27, 2014.
 - [18] A. Hatamlou, "Black hole: a new heuristic optimization approach for data clustering," *Information Sciences*, vol. 222, pp. 175–184, 2013.
 - [19] P. Dłotko and R. Specogna, "Physics inspired algorithms for (co) homology computations of three-dimensional combinatorial manifolds with boundary," *Computer Physics Communications*, vol. 184, no. 10, pp. 2257–2266, 2013.
 - [20] A. Kaveh and M. Khayatad, "A new meta-heuristic method: ray optimization," *Computers and Structures*, vol. 112–113, pp. 283–294, 2012.
 - [21] H. Eskandar, A. Sadollah, A. Bahreininejad, and M. Hamdi, "Water cycle algorithm—a novel metaheuristic optimization method for solving constrained engineering optimization problems," *Computers and Structures*, vol. 110–111, pp. 151–166, 2012.
 - [22] H. Shah-Hosseini, "Otsu's criterion-based multilevel thresholding by a nature-inspired metaheuristic called Galaxy-based Search Algorithm," pp. 383–388, NaBIC, Salamanca, Spain, October 2011.
 - [23] J. J. Flores, R. López, and J. Barrera, "Gravitational interactions optimization," in *Proceedings of the International Conference on Learning and Intelligent Optimization*, pp. 226–237, Springer, Rome, Italy, January 2011.
 - [24] K. Tamura and K. Yasuda, "Primary study of spiral dynamics inspired optimization," *IEEE Transactions on Electrical and Electronic Engineering*, vol. 6, no. S1, pp. S98–S100, 2011.
 - [25] T. T. Hieu, *A Water flow Algorithm for Optimization Problems*, National University of Singapore, Singapore, 2011.
 - [26] A. Kaveh and S. Talatahari, "A novel heuristic optimization method: charged system search," *Acta Mechanica*, vol. 213, no. 3–4, pp. 267–289, 2010.
 - [27] M. Zheng, G.-X. Liu, C.-G. Zhou, Y.-C. Liang, and Y. Wang, "Gravitation field algorithm and its application in gene cluster," *Algorithms for Molecular Biology*, vol. 5, no. 1, p. 32, 2010.
 - [28] L. Xie, J. Zeng, and Z. Cui, "General framework of artificial physics optimization algorithm," in *Proceedings of the World Congress on Nature & Biologically Inspired Computing NaBIC 2009*, pp. 1321–1326, IEEE, Coimbatore, India, December 2009.
 - [29] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: a gravitational search algorithm," *Information Sciences*, vol. 179, no. 13, pp. 2232–2248, 2009.
 - [30] H. Shah-Hosseini, "The intelligent water drops algorithm: a nature-inspired swarm-based optimization algorithm," *International Journal of Bio-inspired computation*, vol. 1, no. 1–2, pp. 71–79, 2009.
 - [31] J. Shen and Y. Li, "Light ray optimization and its parameter analysis," in *Proceedings of the International Joint Conference on Computational Sciences and Optimization CSO 2009*, vol. 2, pp. 918–922, IEEE, Sanya, China, April 2009.
 - [32] M. Kripka and R. M. L. Kripka, "Big crunch optimization method," in *Proceedings of the International Conference on Engineering Optimization*, pp. 1–5, Rio de Janeiro, Brazil, June 2008.
 - [33] M.-H. Tayarani-N and M. Akbarzadeh-T, "Magnetic optimization algorithms a new synthesis," in *Proceedings of the IEEE Congress on Evolutionary Computation CEC 2008 (IEEE World Congress on Computational Intelligence)*, pp. 2659–2664, IEEE, Hong Kong, China, June 2008.
 - [34] R. A. Formato, "Central force optimization: a new meta-heuristic with applications in applied electromagnetics," *Progress In Electromagnetics Research*, vol. 77, pp. 425–491, 2007.
 - [35] C.-L. Chuang and J.-A. Jiang, "Integrated radiation optimization: inspired by the gravitational radiation in the curvature of space-time," in *Proceedings of the IEEE Congress on Evolutionary Computation CEC 2007*, pp. 3157–3164, IEEE, Singapore, September 2007.
 - [36] P. Rabanal, I. Rodríguez, and F. Rubio, "Using river formation dynamics to design heuristic algorithms," in *Proceedings of the International Conference on Unconventional Computation*, pp. 163–177, Springer, Kingston, Canada, August 2007.
 - [37] O. K. Erol and I. Eksin, "A new optimization method: big Bang-Big Crunch," *Advances in Engineering Software*, vol. 37, no. 2, pp. 106–111, 2006.
 - [38] W. F. Sacco and C. R. De Oliveira, "A new stochastic optimization algorithm based on a particle collision meta-heuristic," in *Proceedings of the 6th WCSMO*, Rio de Janeiro, Brazil, June 2005.
 - [39] Y.-T. Hsiao, C.-L. Chuang, J.-A. Jiang, and C.-C. Chien, "A novel optimization algorithm: space gravitational optimization," in *Proceedings of the 2005 IEEE International Conference on Systems, Man and Cybernetics*, pp. 2323–2328, SMC, Waikoloa, HI, USA, October 2005.
 - [40] S. İ. Birbil and S.-C. Fang, "An electromagnetism-like mechanism for global optimization," *Journal of Global Optimization*, vol. 25, no. 3, pp. 263–282, 2003.
 - [41] G. Zarand, F. Pazmandi, K. Pál, and G. Zimányi, "Using hysteresis for optimization," *Physical Review Letters*, vol. 89, no. 15, article 150201, 2002.
 - [42] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: harmony search," *Simulation*, vol. 76, no. 2, pp. 60–68, 2001.
 - [43] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
 - [44] A. Yurtkuran and İ. Kucukoglu, "Comparative study of physics-inspired meta-heuristic algorithms for the solar cell parameter identification problem," in *Proceedings of the 16th International Conference on Clean Energy (ICCE-2018)*, Famagusta, North Cyprus, May 2018.

- [45] H. R. E. H. Boucekara, M. Zellagui, and M. A. Abido, "Optimal coordination of directional overcurrent relays using a modified electromagnetic field optimization algorithm," *Applied Soft Computing*, vol. 54, pp. 267–283, 2017.
- [46] B. Talebi and M. N. Dehkordi, "Sensitive association rules hiding using electromagnetic field optimization algorithm," *Expert Systems with Applications*, vol. 114, pp. 155–172, 2018.
- [47] A. Yurtkuran and E. Emel, "An enhanced artificial bee colony algorithm with solution acceptance rule and probabilistic multisearch," *Computational Intelligence and Neuroscience*, vol. 2016, Article ID 8085953, 13 pages, 2016.
- [48] W.-F. Gao, S.-Y. Liu, and L.-L. Huang, "Enhancing artificial bee colony algorithm using more information-based search equations," *Information Sciences*, vol. 270, pp. 112–133, 2014.
- [49] W. Gong, Z. Cai, and C. X. Ling, "DE/BBO: DE/BBO: a hybrid differential evolution with biogeography-based optimization for global numerical optimization," *Soft Computing*, vol. 15, no. 4, pp. 645–665, 2010.
- [50] D. H. Kim, A. Abraham, and J. H. Cho, "A hybrid genetic algorithm and bacterial foraging approach for global optimization," *Information Sciences*, vol. 177, no. 18, pp. 3918–3937, 2007.
- [51] J. Xiao and L. Li, "A hybrid ant colony optimization for continuous domains," *Expert Systems with Applications*, vol. 38, no. 9, pp. 11072–11077, 2011.
- [52] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 646–657, 2006.
- [53] Y.-T. Juang, S.-L. Tung, and H.-C. Chiu, "Adaptive fuzzy particle swarm optimization for global optimization of multimodal functions," *Information Sciences*, vol. 181, no. 20, pp. 4539–4549, 2011.
- [54] Z.-H. Zhan, J. Zhang, Y. Li, and H. S.-H. Chung, "Adaptive particle swarm optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 6, pp. 1362–1381, 2009.
- [55] S.-C. Chu, J. F. Roddick, and J.-S. Pan, "Ant colony system with communication strategies," *Information sciences*, vol. 167, no. 1-4, pp. 63–76, 2004.
- [56] S. Das, A. Abraham, U. K. Chakraborty, and A. Konar, "Differential evolution using a neighborhood-based mutation operator," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 526–553, 2009.
- [57] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, 2006.
- [58] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.
- [59] K. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*, Springer Science & Business Media, Berlin, Germany, 2006.
- [60] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the Sixth International Symposium on Micro Machine and Human Science MHS'95*, pp. 39–43, IEEE, Nagoya, Japan, October 1995.
- [61] H. Wang, Z. Wu, S. Rahnamayan, H. Sun, Y. Liu, and J.-S. Pan, "Multi-strategy ensemble artificial bee colony algorithm," *Information Sciences*, vol. 279, pp. 587–603, 2014.
- [62] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2009.
- [63] R. Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm: simpler, maybe better," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 204–210, 2004.
- [64] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 240–255, 2004.
- [65] J. Zhang and A. C. Sanderson, "JADE: adaptive differential evolution with optional external archive," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945–958, 2009.
- [66] X. Zhang, X. Wang, Q. Kang, and J. Cheng, "Differential mutation and novel social learning particle swarm optimization algorithm," *Information Sciences*, vol. 480, pp. 109–129, 2019.
- [67] C.-F. Wang and W.-X. Song, "A novel firefly algorithm based on gender difference and its convergence," *Applied Soft Computing*, vol. 80, pp. 107–124, 2019.
- [68] L. Sun, S. Chen, J. Xu, and Y. Tian, "Improved monarch butterfly optimization algorithm based on opposition-based learning and random local perturbation," *Complexity*, vol. 2019, Article ID 4182148, 20 pages, 2019.
- [69] A. A. Heidari, I. Aljarah, H. Faris, H. Chen, J. Luo, and S. Mirjalili, "An enhanced associative learning-based exploratory whale optimizer for global optimization," *Neural Computing and Applications*, pp. 1–27, 2019.
- [70] J. Ding, Q. Wang, Q. Zhang, Q. Ye, and Y. Ma, "A hybrid particle swarm optimization-cuckoo search algorithm and its engineering applications," *Mathematical Problems in Engineering*, vol. 2019, Article ID 5213759, 12 pages, 2019.
- [71] S. Arora and S. Singh, "Butterfly optimization algorithm: a novel approach for global optimization," *Soft Computing*, vol. 23, no. 3, pp. 715–734, 2019.
- [72] A. K. Das and D. K. Pratihari, "A directional crossover (DX) operator for real parameter optimization using genetic algorithm," *Applied Intelligence*, vol. 49, no. 5, pp. 1841–1865, 2019.
- [73] M. López-Ibáñez, J. Dubois-Lacoste, L. Pérez Cáceres, M. Birattari, and T. Stützle, "The irace package: iterated racing for automatic algorithm configuration," *Operations Research Perspectives*, vol. 3, pp. 43–58, 2016.
- [74] B. Akay and D. Karaboga, "Parameter tuning for the artificial bee colony algorithm," in *Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems*, pp. 608–619, Springer, Berlin, Germany, 2009.
- [75] J. Rönkkönen, S. Kukkonen, and K. V. Price, "Real-parameter optimization with differential evolution," in *Proceedings of the Congress on Evolutionary Computation*, pp. 506–513, Edinburgh, UK, September 2005.