A Comparative Study of the Coulomb's and Franklin's Laws Inspired Algorithm (CFA) with Modern Evolutionary Algorithms for Numerical Optimization

Mojtaba Ghasemi[1], Mohsen Zare[2], Amir Zahedi[3], Rasul Hemmati[4], Laith Abualigah[5,6,7,8,9](✉), and Agostino Forestiero[10]

[1] Department of Electronics and Electrical Engineering, Shiraz University of Technology, Shiraz, Iran
[2] Department of Electrical Engineering, Faculty of Engineering, Jahrom University, Jahrom, Fras, Iran
[3] Department of Electrical and Computer Engineering, Tarbiat Modares University, Tehran, Iran
[4] Department of Electrical and Computer Engineering, Marquette University, Milwaukee, Wisconsin, USA
[5] Hourani Center for Applied Scientific Research, Al-Ahliyya Amman University, Amman, Jordan
[6] Faculty of Information Technology, Middle East University, Amman 11831, Jordan
[7] Applied Science Research Center, Applied Science Private University, Amman 11931, Jordan
[8] School of Computer Sciences, Universiti Sains Malaysia, 11800 George Town, Pulau Pinang, Malaysia
[9] Computer Science Department, Prince Hussein Bin Abdullah Faculty for Information Technology, Al Al-Bayt University, Mafraq 25113, Jordan
[10] Institute for High Performance Computing and Networking, National Research Council of Italy, Rende, Italy

Abstract. Coulomb and Franklin's electricity laws are used in this paper to model an efficient optimization algorithm based on electric particle searches, which has been named CFA. For the CFA optimizer, the influence of electrically charged particles on each other in charged things has been predicated on the forces of attraction and repulsion. Evolutionary algorithms (EA) such as hybrid real coded genetic algorithm (RCGA) which combines the global and local search (GL-25), differential evolution (DE) with strategy adaptation (SaDE), composite DE (CoDE), the improved standard particle swarm optimization 2011 (SPSO2013) and the grouped comprehensive learning PSO (GCLPSO) are compared to the CFA optimizer for finding global solutions of seven basic benchmark functions of high dimension D = 50. (GCLPSO). Experiments have shown that the suggested CFA optimizer is quite effective and competitive for the benchmark functions. Note that the source code of the CFA algorithm is publicly available at https://www.optim-app.com/projects/cfa, https://www.mathworks.com/matlabcentral/fileexchange/127727-franklin-s-laws-inspired-algorithm-cfa.

Keywords: Evolutionary algorithms (EAs) · CFA optimizer · Coulomb's and Franklin's laws · high-dimension group search · global numerical optimization

C. Comito and D. Talia (Eds.): PerSOM 2022, LNICST 494, pp. 111–124, 2023.
https://doi.org/10.1007/978-3-031-31469-8_8

1 Introduction

Recent years have seen the use of Physics-Inspired Algorithms (PIAs), such as Atom Search Optimization (ASO) [1] and Wind Driven Optimization (WDO) [2], to address optimization problems that are challenging in the real world, such as non-linearity, non-smoothness, non-convexity, mixed-integer nature, cubic, and non-differentiability. Two major types of optimization techniques are mathematical computing and evolutionary algorithms. These include quadratic programming, linear programming, direct local search methods [3], Nelder and Mead [4], and trust-region quadratic-based models [5]. However, classical mathematical programming approaches cannot provide practical solutions for various optimization problems due to their complexity.

The second group of proposed techniques are meta-heuristic optimization algorithms, which are inspired and modeled by natural phenomena like collective bird and animal behaviors. Examples include Genetic Algorithm (GA) [6], Parti-cle Swarm Optimization (PSO) [7], Differential Evolution (DE) [8], Honey Bees Optimization (MBO) [9], Bacteria Foraging Optimization (BFO) [10], Harmony Search (HS) [11], Cat Swarm Optimization (CSO) [12], Imperialist Competitive Algorithm (ICA) [13], Artificial Bee Colony (ABC) [14], Biogeography-Based Optimization (BBO) [15], Cuckoo Search (CS) [16], Group Search Optimizer (GSO) [17], Chemical Reaction Optimization (CRO) [18], Teaching–Learning-Based Optimization (TLBO) [19], Grey Wolf Optimizer (GWO) [20], and Ant Colony Optimization (ACO) [21], which were suggested and used to solve global numerical optimization issues. These methods have gained popularity in various fields such as production management, power systems, industrial engineering, engineering design, applied mathematics, etc. due to their ease of use and good performance in providing global optimum or near-optimal solutions.

Optimizing complex control variables with several constraints is common in many real-world applications. With rising dimensions and complexity, the global optimization performance of population-based algorithms in such situations tends to decline [22]. Other related methods can be found in [30–37].

This article introduces and develops the CFA optimizer, which is inspired by Coulomb's and Franklin's laws, for optimal search using electric particle population models. The CFA optimizer is both feasible and efficient in addressing optimization issues. The usefulness and accuracy of the CFA optimizer in discovering global solutions to well-known test functions are evaluated and compared to those of other powerful computational with the goal of clarifying the CFA optimizer's core notable features.

2 CFA Optimizer

Coulomb's and Franklin's laws for collective search based on the CFA optimizer are modelled in this section, and the mathematical model for its parameter adjustment is offered.

2.1 Coulomb's and Franklin's Laws

Electric charges are elementary particles that can be classified as positive or negative, as first introduced by Franklin [23]. Typically, objects have an equal balance of positive and negative charges, but charging an object disrupts this balance. When two charged objects, i and j, are brought close together (with charges qi and qj respectively), they exert an attractive or repulsive force on each other, known as electrostatic force. This force is determined by Coulomb's law, which states that the force between two objects is proportional to the product of their charges and inversely proportional to the square of the distance between them. As it can be seen in Fig. 1, Coulomb's law also includes a vector direction, represented by, that is perpendicular to the line connecting the two objects.

This work was driven by the idea that certain individuals in a population can have a positive or negative impact on the development of others. Using the example of a population of four members, A, B, C, and D, where A is the best and D is the worst, members C and D have a negative influence on B, while member A has a positive influence. The reverse is true for D, where all members except D have a negative influence, and for A, where all members except A have a positive influence. The idea behind this is inspired by Coulomb's law, which states that an individual is attracted to those who have a positive impact on their life and repelled by those who have a negative impact, meaning that the positions of those with a positive impact are added to the individual, while the positions of those with a negative impact are subtracted.

2.2 Maintaining Assumptions Based on Coulomb's Law (Attraction and Repulsion) for Optimization

- First assumption: free and moving point charges

The normal state of objects is characterized by an equal number of positive and negative electrical charges. In this state, objects typically consist of Nob distinct elements, each containing n free and mobile point charges, as is the case in gaseous states. Each point charge, represented by q, is made up of D elementary charges (e) or quantized charges. The ith object (Obi) is assumed to have n point charges, and each point charge is considered a potential solution to a problem, with D representing the number of variables to choose from.

- Second assumption: a sign of point charges

In general, objects are made up of an equal number of positive and negative charges. When considering a population of N distinct objects, each with n free, mobile point charges (such as in the gaseous state), each point charge (q) can be viewed as a potential solution to a problem, with D being the number of decision variables. For a specific point charge (qj) of an object (Obi), other point charges within the same object that have a higher objective function value will repel it. In comparison, those with a lower objective function value will attract it. This concept can be illustrated by considering a population of four individuals, where one individual (A) may be positively influenced by one member (B) and negatively influenced by others (C and D).Third assumption: Probabilistic Ionization.

Ionization is the process of separating an electron from the nucleus due to a lack of electrostatic attraction. The energy required to accomplish this is known as ionization

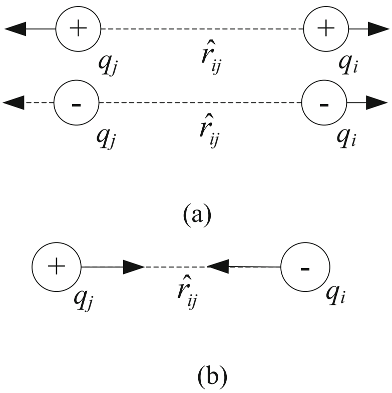

Fig. 1. Coulomb's law, a) Electric repulsion force for two charges with the same signs, b) Electric attraction force for two charges with the opposite signs.

energy. In this context, we consider the possibility that the elementary charge of an object, can be replaced with a new elementary charge under the influence of other charges within the same object.

- Fourth assumption: the probabilistic contact of charged objects to each other

v each one exchanges its highest and lowest ranking point charges with the object to its right and receives the highest and lowest ranking point charges from the object to its left. It's important to note that the point charges exchanged may not necessarily become the highest and lowest ranking point charges for the receiving object.

2.3 CFA Optimizer: The Proposed Algorithm

The working procedure of the CFA optimizer is described in this subsection, which is based on motivated mathematical equations of Coulomb's and Franklin's laws. The steps of the CFA optimizer are as follows:

- **Step 1: Initial population**

At the beginning, N initial solutions are randomly generated as a population in the form of N vectors with D dimensions, with each kth element of the vectors within the lower and upper limits of the kth decision variable ([x_k^{min}, x_k^{max}]). x_j^i The value of x_j^i is determined within the problem's search area using a uniform random variable $U[0, 1]$ as follows:

$$x_{j,k}^{i} = x_k^{\min} + \mathrm{U}[0, 1] \times \left(x_k^{\max} - x_k^{\min}\right), \text{ for } k = 1, \cdots, D \tag{1}$$

- **Step 2: Objects selection for** $Nob \geq 2$

When there are multiple objects, the initial population is organized in order of best to worst objective function values and then divided into Nob groups, each group representing a different object. The population is divided by assigning the first member of the arranged population to the first group, the second member to the second group, and so on until all members are assigned to the groups.

- **Step 3: The attraction/repulsion phase**

In this section, we first sort the members or point charges of the ith group or object based on their objective function value. To calculate the new position of the jth member, xj, of the ith group or object, we randomly select alpha/r members with objective function values less or greater than xj for attraction or repulsion, respectively. We then calculate the average of these selected members, which will be used to compute the new position of xj using a mathematical model based on attraction and repulsion. If the objective function value of the new position is better than the old position, it will become the new position of xj, otherwise, xj will retain its previous position.

$$x_j^{\text{new}} = x_j^{\text{old}} + \left|\cos\theta_j^{\text{new}}\right|^2 \times \left(x^{\text{Best}} - x^{\text{Worst}}\right) + \left|\sin\theta_j^{\text{new}}\right|^2 \times \left(\text{mean}\left(\sum_{m=1\in\ \text{Ob}_i}^{\alpha,\alpha\le\alpha_{\max}} x_m^{\text{Better - than - x}_j}\right) - \text{mean}\left(\sum_{m=1\in\ \text{Ob}_i}^{r,r\le r_{\max}} x_m^{\text{Worse - than - x}_j}\right)\right) \tag{2}$$

In this section, the process of determining the new position of the jth member (xj) of the ith group (Obi) is described using a mathematical model that takes into account attraction and repulsion forces. First, the members of the ith group are sorted based on the value of their objective function. Then, a random number of members with objective function values that are less or greater than xj are chosen for the attraction and repulsion phases, respectively. The average of these chosen members is calculated and used to update the position of xj using an equation. If the objective function value of the new position is better than the old position, xj's position is updated, otherwise it remains the same. The maximum number of negative and positive charges used in averaging (rmax and rmin) are calculated using equations and are affected by a random variable and initial values that are set to be equal to each other. The effect of this process is demonstrated through an example of charging and discharging based on variations of a certain parameter.

$$\begin{aligned} \theta_j^{new} &= \theta_j^{old} + U\left(0,\ \frac{3}{2}\pi\right), \\ \theta_j^{initial} &= U(0,\ 2\pi). \end{aligned} \tag{3}$$

$$\alpha_{\max} = \alpha_0 * (1 + \cos\theta), r_{\max} = r_0 * (1 - \cos\theta), \alpha_0 = r_0. \tag{4}$$

- **Step 4: Probabilistic ionization phase**

In this phase, the ionization process is performed individually for each member, or point charge, of the group. Only the kth decision variable (xj,k) of the member xj of the population is affected by ionization. If the probability value of the normalized ionization energy Pi for the jth member (xj) is greater than a random value rand, the decision variable xj,k is chosen at random using the equation where D is the number of variables in the optimization problem. The new decision variable is obtained using the kth decision variables of the best member and the worst member of the same group according to Eq. (5).

$$\begin{aligned} & \textit{if}\ \text{rand}_i \leq Pi \text{ and } k = \text{round}(1 + \text{rand} * (D-1)) \\ & x_{j,k}^{\text{new}} = x_k^{\text{Best}} + x_k^{\text{Worst}} - x_{j,k} \\ & x_j = x_j^{\text{new}} \end{aligned} \tag{5}$$

- **Step 5: Probabilistic contact phase for the multi-object CFA optimizer**

The process of contact between groups of charges, or members, is done probabilistically, meaning it is done based on a certain probability. When multiple groups of charges are used, the contact operation is performed if a randomly generated value is greater than a constant value called Pc, which is the contact probability factor. In the case of large-dimensional problems, this value was found to be 0.5, which produced the best results in most simulations. The formula for the probabilistic contact phase is outlined in Eq. 6.

$$\begin{aligned} & \text{if rand}_{Iter} \leq Pc \text{ and } Nob \geq 2: \\ & \begin{cases} x^{\text{Best}_1} = x^{\text{Best}_{Nob}},\ \ldots, x^{\text{Best}_{i+1}} = x^{\text{Best}_i},\ \ldots,\ x^{\text{Best}_{Nob}} = x^{\text{Best}_{Nob-1}} \\ x^{\text{Worst}_1} = x^{\text{Worst}_{Nob}},\ \ldots, x^{\text{Worst}_{i+1}} = x^{\text{Worst}_i},\ \ldots,\ x^{\text{Worst}_{Nob}} = x^{\text{Worst}_{Nob-1}} \end{cases} \end{aligned} \tag{6}$$

The flowchart of the optimization process of CFA optimizer is shown in Fig. 2.

***Step 1**: Initial population for N charges or N vectors*

$$x_{j,k}^{i} = x_k^{\min} + \mathrm{U}[0,1] \times \left(x_k^{\max} - x_k^{\min}\right), \quad \text{for } k = 1, \cdots, D$$

***Step 2**: Objects selection for* $Nob \geq 2$

Produce Nob groups via the initial population

***Step 3**: The attraction/repulsion phase*

$$x_j^{\text{new}} = x_j^{\text{old}} + \left|\cos\theta_j^{\text{new}}\right|^2 \times \left(x^{\text{Best}} - x^{\text{Worst}}\right) + \left|\sin\theta_j^{\text{new}}\right|^2$$
$$\times \left(\text{mean}\left(\sum_{m=1 \in \text{Ob}_i}^{\alpha, \alpha \leq \alpha_{\max}} x_m^{\text{Better-than-x}_j}\right) - \text{mean}\left(\sum_{m=1 \in \text{Ob}_i}^{r, r \leq r_{\max}} x_m^{\text{Worse-than-x}_j}\right)\right)$$

if $f\left(x_j^{\text{new}}\right) \leq f\left(x_j^{\text{old}}\right)$

$x_j = x_j^{\text{new}}$

end

***Step 4**: Probabilistic ionization phase*

if $\text{rand}_i \leq Pi$ and $k = \text{round}\left(1 + \text{rand} * (D-1)\right)$

$x_{j,k}^{\text{new}} = x_k^{\text{Best}} + x_k^{\text{Worst}} - x_{j,k}$

end

$x_j = x_j^{\text{new}}$

***Step 5**: Probabilistic contact phase for multi-object CFA*

if $\text{rand}_{Iter} \leq Pc$ and $Nob \geq 2$:

$$\begin{cases} x^{\text{Best}_1} = x^{\text{Best}_{Nob}}, \ldots, x^{\text{Best}_{i+1}} = x^{\text{Best}_i}, \ldots, x^{\text{Best}_{Nob}} = x^{\text{Best}_{Nob-1}} \\ x^{\text{Worst}_1} = x^{\text{Worst}_{Nob}}, \ldots, x^{\text{Worst}_{i+1}} = x^{\text{Worst}_i}, \ldots, x^{\text{Worst}_{Nob}} = x^{\text{Worst}_{Nob-1}} \end{cases}$$

end

Fig. 2. Flowchart of the optimization operation of CFA.

3 Experimental Studies

After Sphere (f_1), Rosenbrock (f_2), Rastrigin (f_3), Griewank (f_4), Schwefel-2–21(f_5), Schwefel-1–2 (f_6), and Weierstrass (f_7) [24] are the seven basic test functions that we used to explore and study the performance and robustness of the proposed CFA in different optimization environments. The detailed explanation of the fundamental test functions have been given in Table 1 and in [24].

3.1 Experiment 1: A Competitive Study for Showing CFA Performance

The performance of the CFA optimizer was compared to that of five other advanced optimization algorithms, including hybrid real coded genetic algorithm (RCGA) which combines the global and local search (GL-25) [25] (http://dces.essex.ac.uk/staff/zhang/), DE with strategy adaptation (SaDE) [26] (www.ntu.edu.sg/home/epnsugan), composite DE algorithm (CoDE) [27] (http://dces.essex.ac.uk/staff/zhang/), the improved standard PSO 2011 (SPSO2011) [28] developed by Mahamed G.H. Omran and Maurice Clerc (by Mahamed Omran http://www.particleswarm.info) (SPSO2013) and the heterogeneous comprehensive learning PSO (HCLPSO) with enhanced exploration and exploitation PSO [29] (www.ntu.edu.sg/home/epnsugan) for test on seven basic benchmarks with $D = 50$. The number of function evaluations (FEs) for all algorithms is FEs $= D * 5000$ and the population size selected for CFA is $N = 20$ and so $Nob = 20/5 = 4$. We employed the identical parameters for these five sophisticated algorithms in this experiment study as the original papers. Thirty separate runs of all algorithms are performed on all of the test functions and issues. This signifies that the performance of each optimization algorithm is either lower than, better than, or similar to that of CFA when compared to the results of CFA and the other five advanced algorithms mentioned as "−", "+", and "=" in the last row of all the best results tables.

Using the identical parameters for all algorithms, the CFA optimizer produced the best results after 30 runs, with the Mean and Standard Deviation (Std.) as listed in Table 2. In this Table, the best findings are given in **boldface**, with the best outcomes for each function. A comparison of the six conventional test functions shows that CFA outperforms the other five algorithms tested. On the test function f_2, the CoDE optimization algorithm outperforms the competition, whereas the SaDE optimization method achieved the same top results with CFA on the test function f_3. CFA outperforms the other five advanced algorithms in this study when it comes to optimizing 50-dimensional test functions.

3.2 Experiment 2: Test on the Effect Pi in the CFA Performance

In this section, we investigate the effectiveness of changing control parameter *Pi* from 0.01 to 0.9 in CFA optimizer on the four test functions f_1, f_2, f_3, and f_7.

Table 3 displays the computational results the Mean and the Std. For *Pi* with 0.01, 0.05, 0.25, 0.5, 0.9 values, after 30 runs for the four test functions f_1, f_2, f_3, and f_7, with the conditions as same as the pervious section and Table 2, the best results on each function have been shown in **boldface**. The best results demonstrate that the proposed algorithm shows efficient performance than the other five advanced algorithms for all the different values *Pi*.

Figure 3 also shows the convergence characteristic of the proposed algorithm with different *Pi* values to demonstrate the convergence feature. As shown in the bottom image, which is a down-scale version of the top figure, the algorithm converged relatively quickly to the global optimal path, and the varying *Pi* had no significant effect on its convergence.

The black color is for the Pi of 0.01, the red for 0.05, the pink for 0.1, the blue for 0.25, the green for 0.5 and the yellow for pi equal to 0.9.

f_7: K = 0.01; r = 0.05; m = 0.1, b = 0.25, g = 0.5; y = 0.9.

Table 1. Summary of the selected test functions with fmin = 0.

Test function	Search Range
$f_1 = \sum_{i=1}^{D} x_i^2$	$[-100, 100]D$
$f_2 = \sum_{i=1}^{D-1} (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$	$[-2.048, 2.048]D$
$f_3 = \sum_{i=1}^{D} (x_i^2 - 10\cos(2\pi x_i) + 10)$	$[-5.12, 5.12]D$
$f_4 = \frac{1}{4000}\sum_{i=1}^{D} (x_i - 100)^2 - \prod_{i=1}^{D} \cos\left(\frac{x_i - 100}{\sqrt{i}}\right) + 1$	$[-600, 600]D$
$f_5(x) = \max_{j=1}^{D} \lvert x_j \rvert$	$[-100, 100]D$
$f_6 = \sum_{i=1}^{D}\left(\sum_{j=1}^{i} x_j\right)^2$	$[-100, 100]D$
$f_7 = \sum_{i=1}^{D}\left(\sum_{k=0}^{k\max}\left[a^k \cos\left(2\pi b^k (x_i + 0.5)\right)\right]\right) - D\sum_{k=0}^{k\max}\left[a^k \cos\left(2\pi b^k\right)\right]$, $a = 0.5\ \ b = 3\ \ k\max = 20$	$[-0.50, 0.50]D$

Table 2. Experimental results of all algorithms over 30 independent runs on 7 test functions of 50 variables with 250,000 fes.

F	GL-25	SaDE	CoDE	SPSO2013	HCLPSO	CFA
f_1	5.70E−85 3.12E−84 −	3.67E−44 6.35E−44 −	3.11E−15 3.01E−15 −	9.59E−05 4.92E−06 −	1.84E−18 9.78E−09 −	**0.00E+00** **0.00E+00**
f_2	4.17E+01 8.91E−01 −	4.41E+01 1.56E+00 −	**3.35E+01** **5.04E−01** **+**	4.35E+01 1.05E+00 −	4.14E+01 3.53E−01 −	4.36E+01 9.66E−01
f_3	5.35E+01 1.21E+01 −	**0.00E+00** **0.00E+00** **=**	7.64E+01 6.08E+00 −	4.36E+01 9.34E+00 −	7.75E+00 1.50E+00 −	**0.00E+00** **0.00E+00**
f_4	4.18E−03 7.24E−03 −	4.06E−03 7.82E−03 −	5.83E−15 7.39E−15 −	6.60E−03 5.72E−03 −	1.52E−03 7.03E−03 −	**0.00E+00** **0.00E+00**

(*continued*)

Table 2. (*continued*)

F	GL-25	SaDE	CoDE	SPSO2013	HCLPSO	CFA
f_5	1.15E+01 2.67E+00 −	3.46E−01 6.73E−01 −	1.95E−03 6.11E−04 -	8.27E+00 1.41E+00 −	5.94E−01 8.96E−02 −	**0.00E+00** **0.00E+00**
f_6	7.59E+02 5.22E+02 −	2.14E+00 1.40E+00 −	5.76E−01 4.16E−01 −	1.17E−03 5.48E−04 −	2.61E+01 1.5E+00 −	**0.00E+00** **0.00E+00**
f_7	1.30E+00 5.38E−01 −	1.36E−01 1.91E−01 −	2.11E−04 5.62E−05 −	1.30E+01 1.15E+00 −	8.38E−06 4.40E−06 −	**0.00E+00** **0.00E+00**
$-/+/=$	7/0/0	6/0/1	6/1/0	7/0/0	7/0/0	−

Table 3. The computational results for the different values Pi.

Pi	0.01	0.05	0.1	0.25	0.5	0.9
$\boldsymbol{f_1}$	**0.00E+00** **0.00E+00**	**0.00E+00** **0.00E+00**	**0.00E+00** **0.00E+00**	**0.00E+00** **0.00E+00**	**0.00E+00** **0.00E+00**	**0.00E+00** **0.00E+00**
$\boldsymbol{f_2}$	4.53E+01 2.47E −01	4.40E+01 7.33E− 02	4.42E+01 1.36E −01	4.42E+01 8.20E −02	4.39E+01 2.81E −01	4.40E+01 2.68E −01
$\boldsymbol{f_3}$	**0.00E+00** **0.00E+00**	**0.00E+00** **0.00E+00**	**0.00E+00** **0.00E+00**	**0.00E+00** **0.00E+00**	**0.00E+00** **0.00E+00**	**0.00E+00** **0.00E+00**
$\boldsymbol{f_7}$	**0.00E+00** **0.00E+00**	**0.00E+00** **0.00E+00**	**0.00E+00** **0.00E+00**	**0.00E+00** **0.00E+00**	**0.00E+00** **0.00E+00**	**0.00E+00** **0.00E+00**

3.3 Experiment 3: Test on the Effect Pc in the CFA Performance

In this section with the conditions as same as the pervious section, we investigate the effectiveness of changing control parameter *Pc* from 0.01 to 0.9 in the suggested algorithm on the four test functions f_1, f_2, f_3, and f_7.

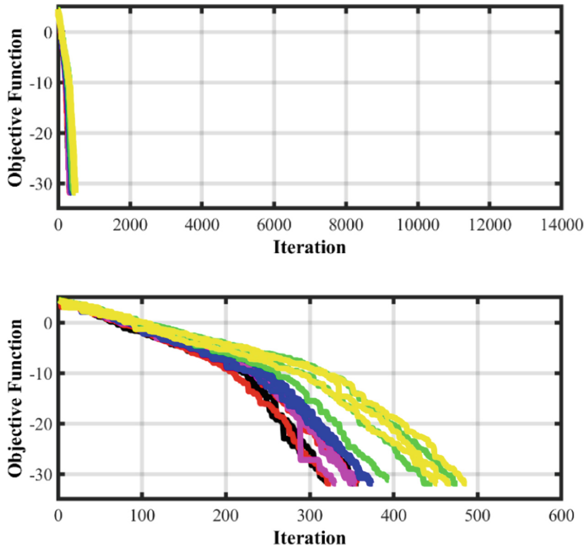

Fig. 3. The convergence graphs of three different runs for each of the *Pi* values for f_7.

Table 4 show the computational results the Mean and the Std. For *Pc* with 0.01, 0.05, 0.25, 0.5, 0.9 values, after 30 runs for the four test functions f_1,f_2,f_3, and f_7. The optimal results illustrate that CFA optimizer shows suitable and noticeable performance than the other five advanced algorithms for all the different values *Pc*.

As in the previous section, to show the CFA convergence characteristic with different Pcs, the convergence characteristic for different values is given in Fig. 4. As can be seen from the figure on the right, which is the down-scale version of the left figure, the CFA quickly converged to the global optimal for this sample test function, and the different Pcs had negligible effect on its convergence.

Table 4. The computational results for the different values Pc.

Pc	0.01	0.05	0.1	0.25	0.5	0.9
f_1	**0.00E+00** **0.00E+00**	**0.00E+00** **0.00E+00**	**0.00E+00** **0.00E+00**	**0.00E+00** **0.00E+00**	**0.00E+00** **0.00E+00**	**0.00E+00** **0.00E+00**
f_2	4.39E+01 6.03E −02	4.40E+01 1.95E −01	4.39E+01 1.69E −01	4.41E+01 8.32E −02	4.42E+01 1.57E −01	4.44E+01 1.74E −01
f_3	**0.00E+00** **0.00E+00**	**0.00E+00** **0.00E+00**	**0.00E+00** **0.00E+00**	**0.00E+00** **0.00E+00**	**0.00E+00** **0.00E+00**	**0.00E+00** **0.00E+00**
f_7	**0.00E+00** **0.00E+00**	**0.00E+00** **0.00E+00**	**0.00E+00** **0.00E+00**	**0.00E+00** **0.00E+00**	**0.00E+00** **0.00E+00**	**0.00E+00** **0.00E+00**

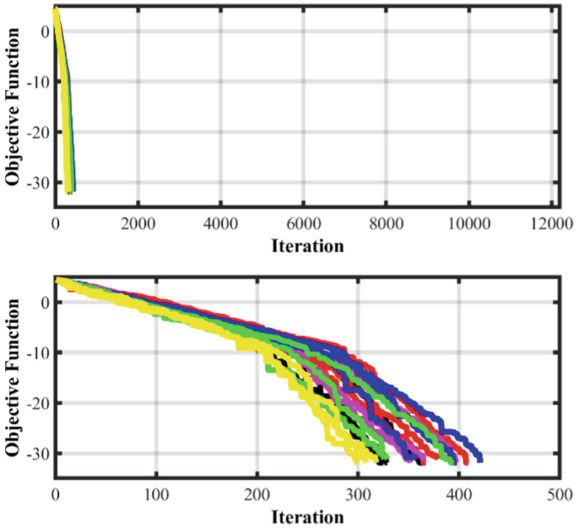

Fig. 4. The convergence graphs of three sample runs for each of the *Pc* values for f_7.

4 Conclusion

Using a new electrostatics model based on Franklin's and Coulomb's laws called CFA, this research investigates how well the CFA optimizer, a novel and notable technique, performs when used to optimize numerical problems using seven well-known fundamental functions. Population-based algorithms like the CFA optimizer can be used to a wide variety of populations, including groups, individuals, and electric particles. The optimization of CFA includes the stages of attraction and repulsion, probabilistic ionization, and probabilistic contact phases. It was shown that CFA was able to outperform five advanced algorithms, i.e. GL-25, SaDE, CoDE, SPSO2011, and HCLPSO, for solving the benchmark functions. Based on our tests, we can conclude that the CFA optimizer we have proposed outperforms the majority of other test functions for high-dimension optimization tasks despite its simple design and straightforward implementation.

References

1. Zhao, W., Wang, L., Zhang, Z.: Atom search optimization and its application to solve a hydrogeologic parameter estimation problem. Knowl. Based Syst **163**, 283–304 (2019)
2. Bayraktar, Z., Komurcu, M., Werner, D.H.: Wind driven optimization (WDO): a novel nature-inspired optimization algorithm and its application to electromagnetics. In: 2010 IEEE Antennas and Propagation Society International Symposium, p. 1–4. (2010)
3. Hooke, R., Jeeves, T.A.: Direct search solution of numerical and statistical problems. J ACM **8**, 212–229 (1961)
4. Nelder, J.A., Mead, R.: A simplex method for function minimization. Comput J **7**, 308–313 (1965)
5. Winfield, D.H.: Function and functional optimization by interpolation in data tables. Harvard University (1970)

6. Mitchell, M.: An Introduction to Genetic Algorithms. MIT press, Cambridge (1998)
7. Eberhart, R., Kennedy, J.: A new optimizer using particle swarm theory. In: In MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, pp. 39-43 (1995)
8. Storn, R., Price, K.: Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim **11**, 341–359 (1997)
9. Abbass, H.A.: MBO: Marriage in honey bees optimization-A haplometrosis polygynous swarming approach. In: Proceedings of the 2001 Congress Evolutionary Computation (IEEE Cat. No. 01TH8546), vol. 1, p. 207–14 (2001)
10. Passino, K.M.: Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Syst Mag **22**, 52–67 (2002)
11. Lee, K.S., Geem, Z.W.: A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Comput. Methods Appl. Mech. Eng. **194**, 3902–3933 (2005)
12. Chu, S.-C., Tsai, P.-W., Pan, J.-S.: Cat swarm optimization. In: Yang, Q., Webb, G. (eds.) PRICAI 2006. LNCS (LNAI), vol. 4099, pp. 854–858. Springer, Heidelberg (2006). https://doi.org/10.1007/978-3-540-36668-3_94
13. Atashpaz-Gargari, E., Lucas, C.: Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In: 2007 IEEE Congress on Evolutionary Computation, p. 4661–7 (2007)
14. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J Glob Optim **39**, 459–471 (2007)
15. Simon, D.: Biogeography-based optimization. IEEE Trans Evol Comput **12**, 702–713 (2008)
16. Yang, X-S., Deb, S.: Cuckoo search via Lévy flights. 2009 World Congr. Nat Biol inspired Comput, p. 210–214 (2009)
17. He, S., Wu, Q.H., Saunders, J.R.: Group search optimizer: an optimization algorithm inspired by animal searching behavior. IEEE Trans. Evol. Comput. **13**, 973–990 (2009)
18. Lam, A.Y.S., Li, V.O.K.: Chemical-reaction-inspired metaheuristic for optimization. IEEE Trans. Evol. Comput. **14**, 381–399 (2009)
19. Rao, R.V., Savsani, V.J., Vakharia, D.P.: Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems. Inf. Sci. **183**, 1–15 (2012)
20. Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. Adv. Eng. Softw. **69**, 46–61 (2014)
21. Drigo, M.: The ant system: optimization by a colony of cooperating agents. IEEE Trans. Syst. Man, Cybern. B **26**, 1–13 (1996)
22. Mahdavi, S., Shiri, M.E., Rahnamayan, S.: Metaheuristics in large-scale global continues optimization: a survey. Inf. Sci. **295**, 407–428 (2015)
23. Halliday D, Resnick R, Walker J. Fundamentals of physics. John Wiley \& Sons; 2013
24. Li, C., Yang, S., Nguyen, T.T.: A self-learning particle swarm optimizer for global optimization problems. IEEE Trans. Syst. Man, Cybern. Part B **42**, 627–646 (2011)
25. García-Martínez, C., Lozano, M., Herrera, F., Molina, D., Sánchez, A.M.: Global and local real-coded genetic algorithms based on parent-centric crossover operators. Eur. J. Oper. Res. **185**, 1088–1113 (2008)
26. Qin, A.K., Huang, V.L., Suganthan, P.N.: Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Trans. Evol. Comput. **13**, 398–417 (2008)
27. Wang, Y., Cai, Z., Zhang, Q.: Differential evolution with composite trial vector generation strategies and control parameters. IEEE Trans. Evol. Comput. **15**, 55–66 (2011)
28. Zambrano-Bigiarini, M., Clerc, M., Rojas, R.: Standard particle swarm optimisation 2011 at cec-2013: A baseline for future pso improvements. In: 2013 IEEE Congress Evolutionary Computation, p. 2337–2344 (2013)

29. Lynn, N., Suganthan, P.N.: Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation. Swarm Evol. Comput. **24**, 11–24 (2015)
30. Gul, F., et al.: A Centralized Strategy for Multi-Agent Exploration. IEEE Access **10**, 126871–126884 (2022)
31. Abualigah, L., Elaziz, M.A., Khodadadi, N., Forestiero, A., Jia, H., Gandomi, A.H.: Aquila optimizer based PSO swarm intelligence for IoT task scheduling application in cloud computing. In: Houssein, E.H., Abd Elaziz, M., Oliva, D., Abualigah, L. (Eds.) Integrating Meta-Heuristics and Machine Learning for Real-World Optimization Problems. Studies in Computational Intelligence, vol. 1038, pp. 481–497. Springer, Cham. (2022). https://doi.org/10.1007/978-3-030-99079-4_19
32. Abualigah, L., Forestiero, A., Elaziz, M.A.: Bio-inspired agents for a distributed NLP-based clustering in smart environments. In: Abraham, A., et al. (eds.) SoCPaR 2021. LNNS, vol. 417, pp. 678–687. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-96302-6_64
33. Alzu'bi, D., et al.: Kidney tumor detection and classification based on deep learning approaches: a new dataset in CT scans. J. Healthc. Eng. 2022 (2022)
34. Khazalah, A., et al.: Image processing identification for sapodilla using convolution neural network (CNN) and transfer learning techniques. In: Abualigah, L. (Eds.) Classification Applications with Deep Learning and Machine Learning Technologies. Studies in Computational Intelligence, vol. 1071, pp. 107–127. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-17576-3_5
35. Melhem, M.K.B., Abualigah, L., Zitar, R.A., Hussien, A.G., Oliva, D.: Comparative study on Arabic text classification: challenges and opportunities. In: Abualigah, L. (eds.) Classification Applications with Deep Learning and Machine Learning Technologies. Studies in Computational Intelligence, vol. 1071, pp. 217–224. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-17576-3_10
36. Anuar, N.A. et al.: Rambutan image classification using various deep learning approaches. In: Abualigah, L. (eds.) Classification Applications with Deep Learning and Machine Learning Technologies. Studies in Computational Intelligence, vol. 1071, pp. 23–43. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-17576-3_2
37. Ke, C. et al.: Mango varieties classification-based optimization with transfer learning and deep learning approaches. In: Abualigah, L. (ed.) Classification Applications with Deep Learning and Machine Learning Technologies. Studies in Computational Intelligence, vol. 1071, pp. 45–65. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-17576-3_3