



Lichtenberg algorithm: A novel hybrid physics-based *meta-heuristic* for global optimization



João Luiz Junho Pereira ^{*}, Matheus Brendon Francisco, Camila Aparecida Diniz, Guilherme Antônio Oliver, Sebastião Simões Cunha Jr, Guilherme Ferreira Gomes

Mechanical Engineering Institute, Federal University of Itajubá (UNIFEI), Itajubá, Brazil

ARTICLE INFO

Keywords:

Lichtenberg Algorithm
Lichtenberg Figures
Lightning
Limited Diffusion Aggregation
Optimization
Metaheuristics

ABSTRACT

This paper proposes a novel global optimization algorithm called Lichtenberg Algorithm (LA), inspired by the Lichtenberg figures patterns. Optimization is an essential tool to minimize or maximize functions, obtaining optimal results on costs, mass, energy, gains, among others. Actual problems may be multimodal, nonlinear, and discontinuous and may not be minimized by classical analytical methods that depend on the gradient. In this context there are metaheuristics algorithms inspired by natural phenomena to optimize real problems. There is no algorithm that is the worst or the best, but more efficient for a given type of problem. Thus, an unprecedented metaheuristic algorithm was created inspired by the physical phenomenon of radial intra-cloud lightning and Lichtenberg figures, successfully exploiting the fractal power and it is different from many in the literature as it is a hybrid algorithm composed of methods of search based on population and trajectory. Several test functions, including a design problem in a welded beam, were used to verify the robustness and to validate the Lichtenberg Algorithm. In all cases, the results were satisfactory when compared to those in the literature. LA shown to be a powerful optimization tool for both unconstraint optimizations and real problems with linear and nonlinear constraints.

1. Introduction

The classic optimization consists of methods that depend on the continuity and gradient of functions, being solved in most cases by analytical and mathematical methods. As in practical engineering problems most applications are normally nonlinear and have complicated or nonexistent analytical solutions, they require sophisticated optimization tools to be determined (Yang, 2014; Gomes & Giovani, 2020).

Some behaviors found in nature are sources of inspiration for the development of numerical optimization algorithms that aim to obtain optimal solutions with lower computational cost and possible operating time (Mirjalili & Lewis, 2016). These types of algorithms generate random solutions in the search space for a given problem and continuously improve them with each iteration, not just randomly, but with tradeoffs that make these solutions converge to possible global optimums. Due to this characteristic, these algorithms are called metaheuristics (Yang, 2014) and each metaheuristic has its own parameters

that regulate its optimization process (Nabil, 2016).

The greatest difficulties of these algorithms are in having a correct balance between exploration and exploitation, this is respectively, being able to escape from local minimums and still being able to improve the precision of the solutions already found (Olorunda & Engelbrecht, 2008). For Yang (2014), among the various algorithms there are no good or bad algorithms, but one more appropriate for a given optimization problem, that is, still have space for the development of new metaheuristic algorithms.

A new metaheuristic inspired by the physical phenomena of lightning storms, which anyone who has observed it can confirm its powerful speed, range and energy, is proposed to attack the search space for optimal solutions for complex optimization problems. A tool to match this inspiration was very important: the Lichtenberg Figures (LF). These are reproductions of radial electrical discharges propagations in environments with electrical resistance (dielectric).

Ten test functions (unimodal and multimodal) and a test problem with constraint will be targets of the new metaheuristic and their results will be fully analyzed. All details of creating a powerful global

* Corresponding author.

E-mail addresses: joaulizjp@gmail.com (J.L.J. Pereira), matheus_brendon@unifei.edu.br (M.B. Francisco), camiladiniz@unifei.edu.br (C.A. Diniz), gaoliver@unifei.edu.br (G. Antônio Oliver), sebas@unifei.edu.br (S.S. Cunha), guilhermefergom@unifei.edu.br (G.F. Gomes).

Nomenclature	
$N_{cluster}$	Number of particles in the cluster (DLA)
$R_{cluster}$	Geometric radius of the cluster (DLA)
S	Stickiness coefficient (DLA and LA)
N_p	Number of particles (LA)
R_c	Creation radius (LA)
Ref	LA refinement
M	Figure switching parameter in LA
Pop	Number of Population used in LA
N_{iter}	Number of LA iterations
WBD	Welded Beam Design
h	Width of the welded area (WDB)
l	Lenght of the welded area (WDB)
t	Debt of the beam area (WDB)
b	Thickness of the beam area (WDB)
E	Longitudinal Elastic Constant
G	Torsional Elastic Constant
d	Number of problem variables
GA	Genetic Algorithm
PSO	Particle Swarm Optimization
APSO	Accelerated PSO
SFS	Stochastic Fractal Search

optimization technique are presented. Although many studies have been reported on the development of new global optimization algorithms, none has been found based on Lichtenberg figures. The work presented here assesses the potential of LA for a robust and real engineering problem.

Recently, the LA was applied in the identification and characterization of crack propagation in thin plates of composite material. The algorithm accurately identified the presence, direction and intensity of propagation of edge and central cracks, containing four and eight complex variables respectively, using a small number of sensors (Pereira et al., 2020).

This manuscript is organized as follows: Section 2 a general theoretical background review is presented. Section 3 presents in detail the development of the Lichtenberg Algorithm. In section 4 are the validation methodologies and the presentation of the test problem, a complex structural optimization problem. Section 5 presents the main numerical results and discussion. Finally, Section 5 draws the conclusions.

2. Backgrounds

2.1. Global optimization

According to Yang (2014), real problems can be composed of functions discontinuous and without derivatives, can have more than one variable and composition of objective functions, can be “multimodal”, which cause classic methods to fall into local minimum or maximum and get stuck, since they are based on gradients.

Metaheuristics are developed to solve sophisticated computing problems with reasonable time and good precision (Chopard & Tomassini, 2018). For Yang (2014), there is a classification for metaheuristic algorithms that can be “based on trajectories”, such as Simulated Annealing - created by Kirkpatrick et al. (1983) and the “population-based” ones that are more popular, such as Genetic Algorithm (GA) - One of the first to be created and was by Holland (1975); Ant Colony (ACO) - created by Marco Dorigo (1992); Particle Swarm (PSO) - developed by Kennedy and Eberhart (1995); Sunflower Optimization (SFO) - developed by Gomes et al. (2019). Still, there may be those based on vectors, such as the Differential Evolutionary (DE) algorithm - created by Storn and Price (1997).

In general, there are four main groups that divide metaheuristics according to the inspiration for their creation: i) based on evolution (nature), ii) based on physical phenomena, iii) based on behaviors related to humans and iv) based on swarms (HEIDARI ET AL., 2019; Gomes & Pereira, 2020). Evolutionary algorithms are inspired by plants and animal behavior, such as GA, DE and SFO. Those based on physical phenomena are inspired by laws of physics, such as heat transfer, gravitational force, particle motion, wave propagation and others as SA; the Gravitational Search Algorithm (GSA) - created by Rashedi et al. (2009); the Heat Transfer Search (HTS) - created by Patel and Savsani (2015); and the Black Hole - created by Hatamlou (2013).

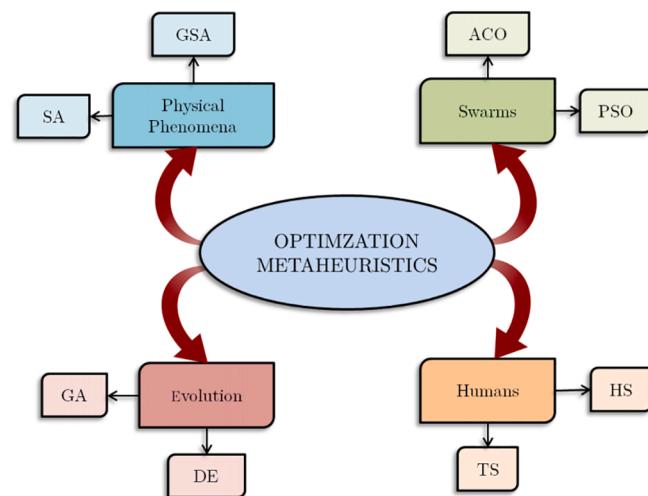


Fig. 1. Diagram relating the metaheuristics, the sources of inspiration and the algorithms created.

The ones based on swarms are those inspired by intelligent behaviors of collective movements seen in colonies and/or societies. These collective behaviors were the inspiration for the ACO; PSO; Whale Optimization Algorithm (WOA) - created by Mirjalili and Lewis (2016), Moth-flame Optimization (MFO) – created by Mirjalili (2015), among others. Finally, we have as examples of algorithms inspired by human behaviors like the Search Harmony (HS) - created by Geem et al., 2001, the Social Emotional Optimization (SEO) - created by Xu et al., 2010 and Political Optimization (PO) – created by Qamar Askari et al. (2020). Fig. 1 summarizes some techniques exposed above.

The drawback of the above cited algorithms is a correct balance between exploration and exploitation (Olorunda & Engelbrecht, 2008; Gomes & de Almeida, 2020). The first refers to the ability to leave local minimum and better explore the entire search space in order to find new solutions. The second refers to localized exploration to improve the accuracy of the solutions that have already been found (Elaziz & Mirjalili, 2019). It is necessary to apply different algorithms for the same case, in order to be able to assess their efficiency in the face of a problem and compare them with each other. There is still the possibility to create new algorithms

2.2. Nature inspiration source

Anyone on rainy days has had the experience of contemplating lightning and observing its propagation speed, its range and especially its ability to branch out by repeating its macrostructures in microstructures while moving between its objectives. Those characteristics

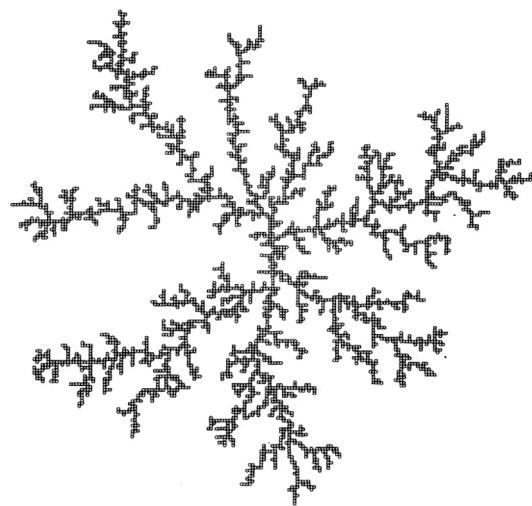


Fig. 2. Agglomerate generated by the diffusion limited aggregation model for 3600 particles (Witten & Sander, 1981).

can be applied in search space in order to find the global optimum in a problem and balancing exploration and exploitation can result in a high-performance metaheuristic. Thus, there is a proposal for a new metaheuristic inspired by the physical phenomenon of lightning storms.

According to Pinto (1987), a storm cloud is produced from water vapor that can be up to 15 km in altitude. As this vapor rises by convection due to the higher temperature close to the earth, it turns into a liquid or solid and can acquire shapes such as drops of water, super-cooled water droplets, snow crystals, light hail, hailstones and ice crystals as a function of altitude, temperature and relative humidity. These structures evolve into a storm cell and then have several cells electrified due to mutual collisions and water frictions in their various states caused by strong upward and downward movements within the cloud. The electric field intensifies and gives rise to lightning or atmospheric charges. There may be lightning from the sky and from the ground. Within the sky, there are intra-cloud lightning (same cloud), inter-cloud (different clouds) and in the air (starts in the cloud and ends in the atmosphere). There are also lightning that relate to the ground, there is the cloud-ground and the ground-cloud. Intra-cloud and cloud-ground are the most common.

The intra-cloud type has a good area scan and in order to create an optimization algorithm that looks for an optimal point in the search space, it is the best type. It is the one that can spread radially, especially when there is one type of charge in the center and the opposite in the peripheries of the cloud. Yet according to Naccarato (2001), discharges that manifest themselves horizontally tend to be more branched.

Lichtenberg (1777) was the first to study this phenomenon of propagation of electric discharges in dielectric (resistant) material - which leads the figure to have these branched and tortuous aspects. According to Merrill and Von Hippel (1939), the impossibility of determining the resistances at each point, having the heterogeneities of the material, determines the random growth of the figure for each case, even if on the same material and electrostatic conditions. It is a stochastic growth, therefore, random. It is observable in these figures the representation of the larger figure in a similar way in increasingly smaller structures. It is the formation of a fractal. Lichtenberg (1777) and Merrill and Von Hippel (1939) did not deal with an algorithm that could generate it randomly. These dealt with physical conditions to create it in existing material.

2.3. Method to build a Lichtenberg Figure

Turner (2019) suggests that LF can be built through a random growth process with many particles, forming a cluster. Because it is a stochastic

model, each execution of the algorithm can generate different figures. Two models of cluster growth have been found in the literature: the Diffusion Limited Aggregation (DLA) and Dielectric Breakdown Model (DBM) (Nienmeyer et al., 1984). Irurzun et al. (2002) carried out a series of studies on the ability to build an electrical tree structure using DBM and DLA and concluded that there are no statistical differences, making the DLA the model of cluster construction chosen for the construction of the Lichtenberg Figure.

This theory was presented by Witten and Sander (1981) as a numerical simulation model for the growth of clusters based on the principle of solid particles. A random structure growth model used to study the formation of figures in the breaking of resistance of a dielectric. The size of the cluster and the number of particles are defined when starting the algorithm, but the shape and size are totally undefined and unique after each execution of the algorithm. According to Meakin and Tolman (1989), this cluster growth model can lead to complex and chaotic behaviors that can often be described in terms of fractal geometry. Without computers it would be impossible. Indeed, several experiments in the literature prove that LF corresponds well to mathematical parameters that are used in fractals. Fig. 2 shows the first cluster simulated in history by the DLA model

Witten and Sander (1983) in addition to proposing these equations, they added a stick coefficient (S). This parameter determines whether a particle with a random walk when encountering the agglomerate is fixed or not. The closer to 1 is S , the more the particles stick to the agglomerate and the faster it grows, which causes the cluster to quickly reach its spatial limits of creation and ends up having a low density.

Witten and Sander (1983) in addition to proposing these equations, they added a stick coefficient (S). This parameter determines whether a particle with a random walk when encountering the agglomerate is fixed or not. The closer to 1 is S , the more the particles stick to the agglomerate and the faster it grows, which causes the cluster to quickly reach its spatial limits of creation and ends up having a low density.

Fractals have a parameter called Fractal Dimension (D), which, contrary to what they think, is not about its size. It is a parameter that statistically measures how much a fractal fills a space. Most of the time is close to 1.7 in the DLA model (Di Rocco & Model, 2009). It can be calculated by Eq. (1), where $N_{cluster}$ is the number of particles and $R_{cluster}$ is the average radius that the cluster occupies:

$$D = \ln(N_{cluster}) / \ln(R_{cluster}) \quad (1)$$

2.4. Similar inspirations for metaheuristics

There are two algorithms that share similar ideas: Stochastic Fractal Search (SFS), created by Salimi (2014) and Lightning Search Algorithm (LSA), created by Shareef et al. (2015). None of these algorithms are composed of search methods based on trajectory and population at the same time. The main idea of the metaheuristic created is to combine these two search methods in order to distribute points to be evaluated in the objective function in regions that sometimes occupy the entire search space and sometimes much smaller regions, generating an excellent balance between exploration and exploitation.

3. Lichtenberg algorithm

The numerical basis for the development of the algorithm was the Diffusion Limited Aggregation (DLA) theory, from which the LF is built following the model presented by Witten and Sander (1981) and Witten and Sander (1983). A binary matrix (0 and 1) is built like a map and in the center a particle, represented by the number one, is fixed. The cluster is built by the values of the matrix that is one and the empty spaces have zero value. Each matrix element of value one is a particle of the cluster and the number of them (N_p) in the cluster is defined at the beginning of the program. This matrix form can be represented by a Bitmap figure (black and white). The space for construction of the figure is defined by

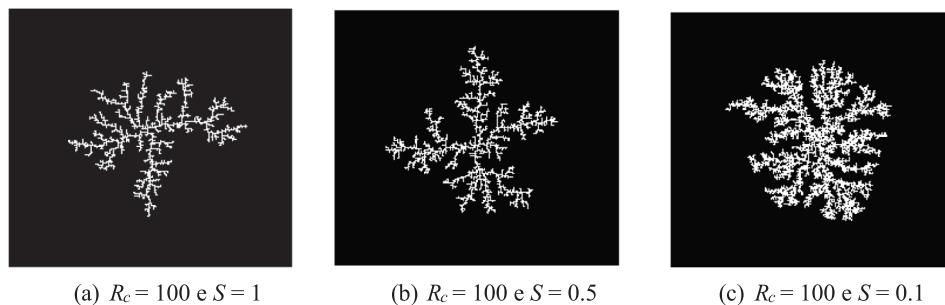


Fig. 3. Bitmapped Lichtenberg figures generated at the beginning of the LA simulation.

Table 1
Results Bitmap figures.

Figure	R_c (input)	N_p (input)	S (input)	$N_{cluster}$	$R_{cluster}$	D
a	100	10,000	1	1851	4.4198	1.6995
b	100	10,000	0.5	2771	3.8834	1.7974
c	100	10,000	0.1	2125	4.5870	1.849

the creation radius (R_c) and from it the matrix is generated with line and column numbers equal to twice R_c (diameter).

Particles are randomly released across the matrix and if they reach the cluster that in the beginning it was just a particle in the center, they have an S probability of fixing, also called the stickiness coefficient. The particle walks being plotted randomly, radially and as if it were on a Cartesian plane from the center and settling down anywhere on the map, rounding off the position to a matrix element with line and column. At this point it can be added only if there is another particle next to it confirmed by a lateral check. If it reaches a radius slightly larger than the R_c , it is exterminated and another one starts the random walk again. This happens until all the particles determined at the entrance N_p are contained in the cluster or until it reaches its limit of construction.

Each execution of the construction of the LF generates totally different figures and the parameter stickiness coefficient (S) influences considerably the density of the figure. As the cluster grows randomly and is finished when the particles reach the creation radius or have already been added to the cluster, the average cluster radius or the number of particles may be different from those ordered at the beginning of the program. In this way, the real number of particles in the cluster is counted ($N_{cluster}$) and its final radius is calculated. These two variables are used to calculate the Fractal Dimension (D).

The results of the programming that generates the FL in bitmap or matrix format are very reliable to those found in the literature, with a D close to 1.7. Three important parameters have been presented and are exclusively for the construction of LF: N_p , S and R_c . Fig. 3 shows LF in Bitmap with different values of S and the same N_p and R_c . Note the difference in density (D) in Table 1. The lower S is, the less likely the particles are to stick to the cluster and the density of the cluster increases. Each particle in the cluster can be transformed into locations on a Cartesian plane and the LF can be plotted at any size, slope or starting point. Then, at first the extracted figure is plotted in the exact size of the search space and its center in the center of it. Fig. 4 illustrates this.

If the LF is plotted exactly like this, there will always be a distance between one point and another and when using it on an optimizer, gaps would exist and the accuracy of the optimizer would be compromised. For this, four measures were taken.

The first is that a random multiplier between 0 and 1 acts at each iteration on the size of the LF, which in its original is the size of the problem search space. That is, at each iteration the figure is plotted in a different size, ranging from the size of the search space to <1% of its size. This greatly reduces the distance between points with each iteration and improves the efficiency of the LA.

A second measure is the addition of random rotation to the LF at each iteration. An angle is randomly given and the entire structure is rotated according to the point rotation equations in Cartesian equations. This always occurs for 2D and 3D figures and when there are >3 variables, the mirroring of these figures is given having different rotations for the figures in the previous planes. This greatly improves the results, as it avoids the simple repetition of the stitches. Fig. 5 illustrates some iterations showing the first and second measures.

The third was the addition of a refinement (ref), an input parameter

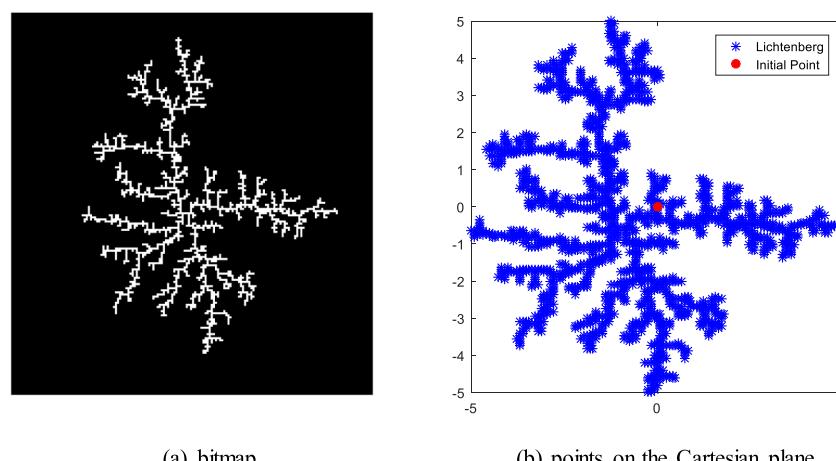


Fig. 4. Transformation of bitmap figures into coordinates plotted in the range $[-5, 5]$.

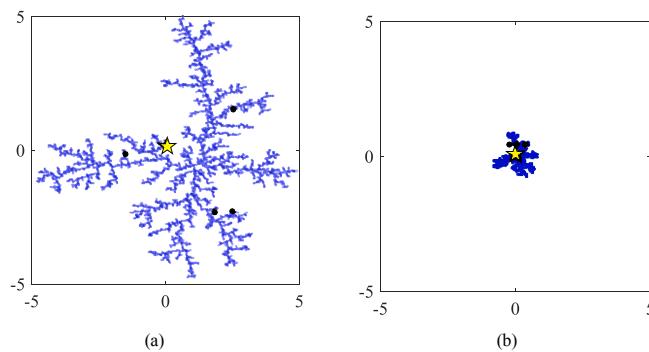


Fig. 5. Exemplification of scales and random rotations for the same Lichtenberg figures.

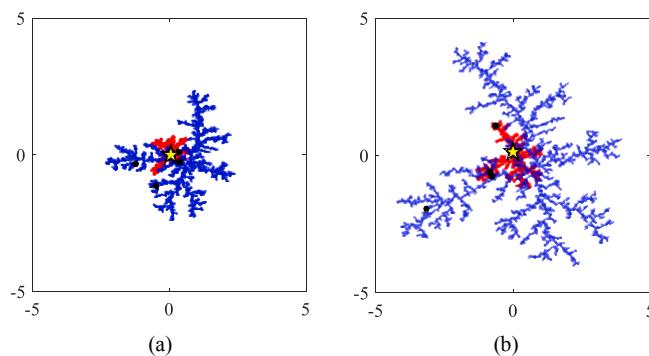


Fig. 6. Local Figure with 30% of the Global size and some iterations.

that can be from 0 to 1 and is a creator of a second LF (red) every iteration from 0 to 100% the size of the main LF (blue) with the same trigger point, rotation and scale factor. This smaller scale figure improves local search. If $ref = 0$, only one LF acts on the optimizer every iteration (blue), the global one. Fig. 6 shows two examples of iterations with $ref = 0.3$. The yellow star is the optimum point of the previous iteration and the LF trigger point of the current iteration.

Not all LF points are used to compute the objective function, the number of points used for this purpose or population (pop) is defined in the beginning of algorithm and is usually 10 times the number of design variables of the problem. The LF points that will represent the population are chosen throughout the LF structure (which is modified at each iteration) and are represented graphically by black dots, all of which are always within the search space by means of a check. This form makes LA a hybrid algorithm as it merges two types of algorithms found in the literature: population and trajectory.

This greatly reinforces the efficiency of the optimizer, since in some

moments, all points of the iteration are required to sweep a region much smaller than the entire search space. While at other times, the figure becomes larger and a much larger space is swept back. Having the secondary LF which is inferior in nature to the primary, the local search is even more efficient.

Various criteria for the distribution of the population between primary and secondary figures have been adopted, however, generally the distribution that shows the best result is 40% of the total pop in LF Global and the rest in Local, as shown in Fig. 7 for 3D and Fig. 8 for 2D. Both are 30% refined and pop equal to 10 just for example.

The fourth measure taken to not discretize the search space is the insertion of an M parameter for changing the LF in the optimizer input data. This parameter can be worth zero, one or two. If zero, a figure is generated when starting the program and the same figure is used in all the iterations of the execution. If it is 1, a new figure is generated at each iteration. If M is worth 2, a previously saved figure is used in the optimizer: no figure is generated.

Finally, the number of iterations (N_{iter}) is also defined as an initial configuration parameter of the algorithm LA, being equal to 100 iterations, a very common number for many algorithms found in the literature. At each iteration, the starting point of the LF trigger is the optimal point of the previous iteration and in the first iteration, it is one of the LF points randomly chosen.

This algorithm is used for the construction of flat (2D) and spatial (3D) figures. As the algorithm needs a physical space to be built, when dealing with more variables, a projection (or mirroring) of these figures is made for n variables. Table 2 shows the recommendations of its parameters, Fig. 9 shows the complete flowchart of the algorithm and Table 3 summarizes the algorithm through a pseudo code.

The LA maps the search space with a limited number of points

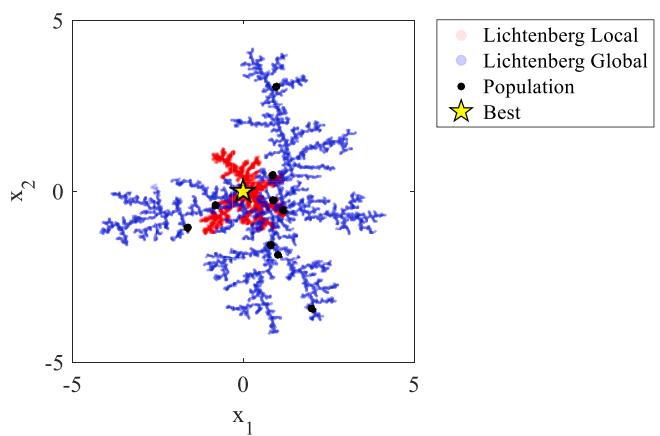


Fig. 8. Population distribution in the Lichtenberg Figures ($pop = 10$ and $ref = 0.3$).

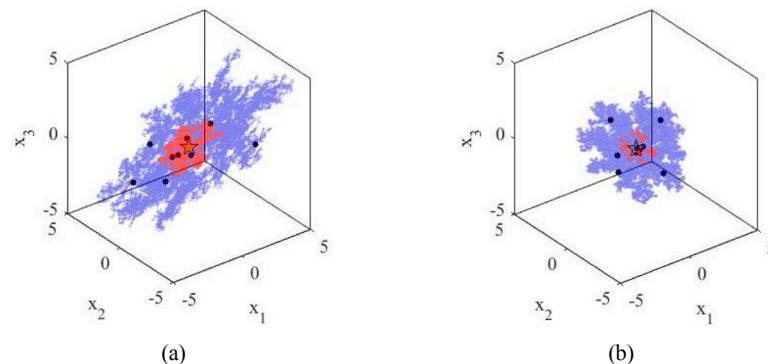


Fig. 7. Population distribution in LA for three variables and two examples of iterations ($pop = 10$ and $ref = 0.3$).

Table 2
Recommended LA Parameters.

Parameters	2D and >3D	3D
R_c	50 to 200	50 to 150
N_p	$>10^3$ and $<10^6$	$>10^5$ and $<10^6$
S	0 to 1	0 to 1
Pop	$(10 \times 40) \times d$	$(10 \times 40) \times d$
ref	0 to 1	0 to 1
M	0, 1 or 2	0 or 1
N_{iter}	$>10^2$ and $<10^3$	$>10^2$ and $<10^3$

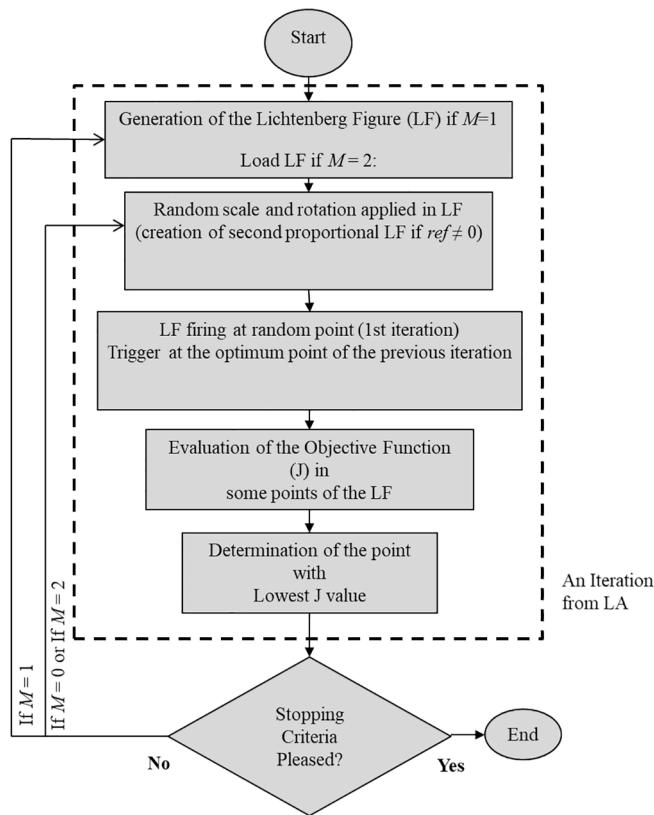


Fig. 9. Lichtenberg Algorithm Flowchart.

defined by the operator in Pop to be evaluated in the objective function, where the best point (lowest value) of the previous iteration is always the trigger point of the current iteration, but it will not necessarily be a point to be evaluated again. All the other Pop points in the current iteration are randomly selected in the LF. When this is plotted with a small random scale, the result of the optimization gains exploitation. In iterations where these figures are plotted on larger random scales, the optimizer exploits its ability to extrapolate local minimums redistributing points to be assessed in regions not yet explored by the Lichtenberg storm. This brought the optimizer the ability to quickly converge with few points of evaluation and iterations.

4. Methodology - applications

Table 4 shows the test functions for nonlinear unconstrained optimization used. The number of variables in which this function was tested, the search space for each variable that are the same and the optimum function values (F_{opt}) are all shown in this table. These functions were presented by Mirjalili and Lewis (2016). The functions from f_1 to f_6 are unimodal and the others multimodal.

The chosen test problem contains non-linear constraints: the welded beam design, where it is necessary work with four variables: width (h)

Table 3
Pseudocode of the Lichtenberg Algorithm.

Algorithm 1 - Main

```

Set objective function and search space –  $J$ , upper and lower bounds
Set number of iterations and population –  $N_{iter}$ ,  $Pop$  (common to all optimizers)
Set Refinement and Parameter for changing the LF –  $Ref$ ,  $M$  (LA routine parameters)
Set LF Parameters –  $R_c$ ,  $N_p$ ,  $S$ 
if  $M = 2$ , load LF, end if
if  $M = 0$ , Create a LF, end if
while (iter <  $N_{iter}$ ) do
    if  $M = 1$ , Create a LF, end if
     $X_{trigger}$  = search space center (trigger point of the first LF)
    if  $ref = 0$ 
        Apply random scale and rotation
        Initialize random population through LF,  $X_i$  ( $i = 1, 2, \dots, Pop$ )
    else
        copy LF to create a second LF of size  $ref * LF$  (Local)
        Apply the same random scale and rotation to both
        Initialize global random population through LF,  $X_{global_i}$  ( $i = 1, 2, \dots, 0.4*Pop$ )
        Initialize local random population through LF,  $X_{local_j}$  ( $j = 1, 2, \dots, 0.6*Pop$ )
         $X_i = X_{global_i} + X_{local_j}$ 
    end if
    Calculate the fitness of each  $X_i$ 
 $X_{best}$  = the lowest  $X_i$  value found
 $X_{trigger} = X_{best}$ 
Iter = iter + 1
end while
return  $X_{best}$ 

```

Algorithm 2 – Creation of LF

```

Create an matrix of  $R_c$  - sized zeros
Place a unitary particle in its center
While ( $i < N_p$ ) do
    Randomly place a unitary particle in the matrix
    if the plotted unitary particle  $t$  is next to another unitary particle
        if rand <  $S$ 
            This new unitary particle is placed in the matrix
             $i = i + 1$ 
        else
            The plotted unitary particle is eliminated
            end if
        end if
    if the cluster of unitary particles reaches  $R_c$ 
        The simulation is finished
        end if
    end while
 $X$  = coordinates of all unitary particles for Cartesian space in the size of the search space.

```

and length (l) of the welded area, the depth (t) and the thickness (b) of the beam. This problem is outlined in Fig. 10. The objective is to minimize the total manufacturing cost under the appropriate constraints of shear stress (τ), bending stress (σ), buckling load (P) and maximum deflection (δ). According to Rao (2009), the mathematical modeling of the problem is represented by Equations 3 and 4 and the appropriated objective function is given by Eq. (2). Where $x_1 = h$, $x_2 = l$, $x_3 = t$ and $x_4 = b$.

$$\min(x) = 1.1047x_1^2x_2 + 0.04811x_3x_4(14 + x_2) \quad (2)$$

Subject to

$$g_1(X) = \tau(X) - \tau_{\max} \leq 0 \quad (3.a)$$

$$g_2(X) = \sigma(X) - \sigma_{\max} \leq 0 \quad (3.b)$$

$$g_3(X) = x_1 - x_4 \leq 0 \quad (3.c)$$

$$g_4(X) = 0.10471x_1^2 + 0.04811x_3x_4(14 + x_2) - 5.0 \leq 0 \quad (3.d)$$

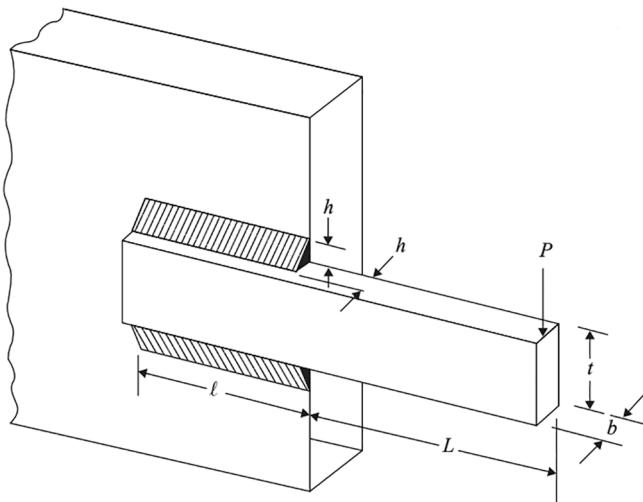
$$g_5(X) = 0.125 - x_1 \leq 0 \quad (3.e)$$

$$g_6(X) = \delta(X) - \delta_{\max} \leq 0 \quad (3.f)$$

Table 4

Test Functions for LA Validation.

Test Function	Number of variables	Search Space	F_{opt}
$f_1 = \sum_{i=1}^n (x_i^2)$	1, 2, 3, 4, 5	[-100, 100]	0
$f_2 = \sum_{i=1}^{n-1} 100(x_{i+1} - x_i)^2 + (x_i - 1)^2$	2, 3, 4, 5	[-10, 10]	0
$f_3 = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i$	2	[-10, 10]	0
$f_4 = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	2	[-10, 10]	0
$f_5 = \sum_{i=1}^n (x_i + 0.5)^2$	2	[-10, 10]	0
$f_6 = \sum_{i=1}^n i x_i^4 + \text{rand}[0, 1]$	2	[-1.28, 1.28]	0
$f_7 = 20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)) = 20 + e$	2	[-32, 32]	0
$f_8 = \sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i) + 10]$	2	[-5.12, 5.12]	0
$f_9 = 4x_1^2 - 2.1x_1^4 + 0.3333x_1^2 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[-5, 5]	-1.0316
$f_{10} = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6 \right)^2 + 10\left(1 - \frac{1}{8\pi} \right)\cos x_1 + 10$	2	[-5, 5]	0.398

**Fig. 10.** The welded beam problem (Rao, 2009).

$$g_7(X) = P - P_C(X) \leq 0 \quad (3.g)$$

$$0.1 \leq x_i \leq 2.0, i = 1, 4 \quad (3.h)$$

$$0.1 \leq x_i \leq 10.0, i = 2, 3 \quad (3.i)$$

where:

$$\tau(X) = \sqrt{(\tau')^2 + 2\tau' \frac{x_2}{2R} + (\tau'')^2} \quad (4.a)$$

$$(3.g)$$

$$(3.h)$$

$$(3.i)$$

The physical and geometrical parameters of the welded beam are: Load (P) = 6000 lb; Beam Length (L) = 14 in; Longitudinal Elastic Constant (E) = 30×10^6 psi; Torsional Elastic Constant (G) = 12×10^6 psi; Maximum Shear Stress (τ_{max}) = 30000 psi and Maximum Deflection (δ_{max}) = 0.25 in.

5. Numerical results and discussion

5.1. Test functions

Each function in Table 3 was simulated 30 times and the parameters

Table 5

Optimization results of two-dimensional test functions.

Function	x_1		x_2		f_{min}		Target (F_{opt})
	Mean	SD	Mean	SD	Mean	SD	
f_1	-0.0015	0.0277	0.0117	0.0457	0.0029	0.0044	0
f_2	1.0327	0.2090	1.1091	0.6400	0.0454	0.2267	0
f_3	-0.0001	0.0032	0.0001	0.0037	0.0047	0.0038	0
f_4	-0.0020	0.0033	0.0024	0.0049	0.0000	0.0000	0
f_5	-0.5001	0.0032	0.5000	0.0046	0.0000	0.0000	0
f_6	0.0109	0.1348	-0.0216	0.1022	0.0020	0.0016	0
f_7	-0.0018	0.0084	0.0024	0.0113	0.0377	0.0327	0
f_8	0.0003	0.5537	0.0028	0.2519	0.0016	0.4850	0
f_9	-0.01782	0.0891	0.1416	0.7097	-1.0318	1.0897	-1.0316
f_{10}	3.1418	0.0011	2.2742	0.0028	0.3979	0.0000	0.398

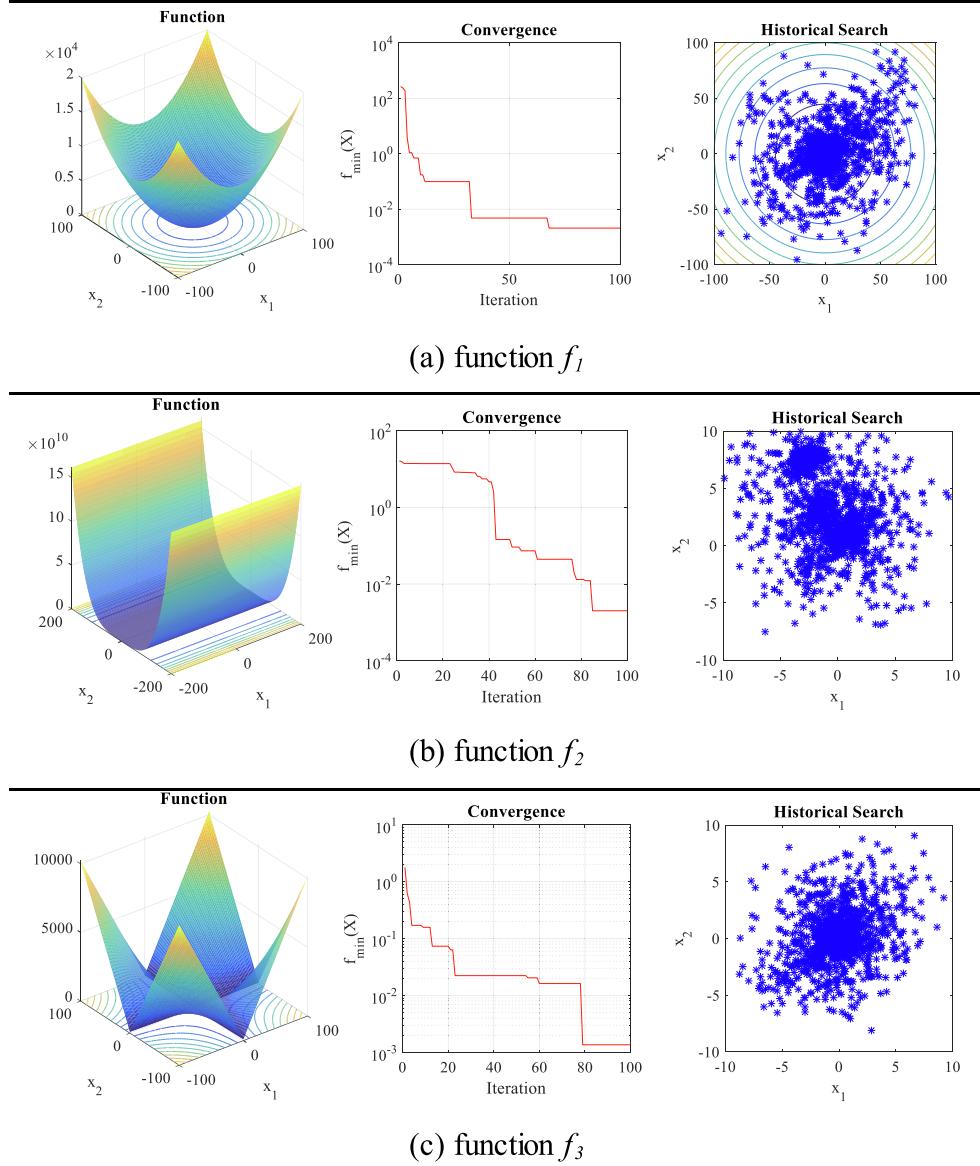


Fig. 11. Analysis of search results for two-dimensional functions.

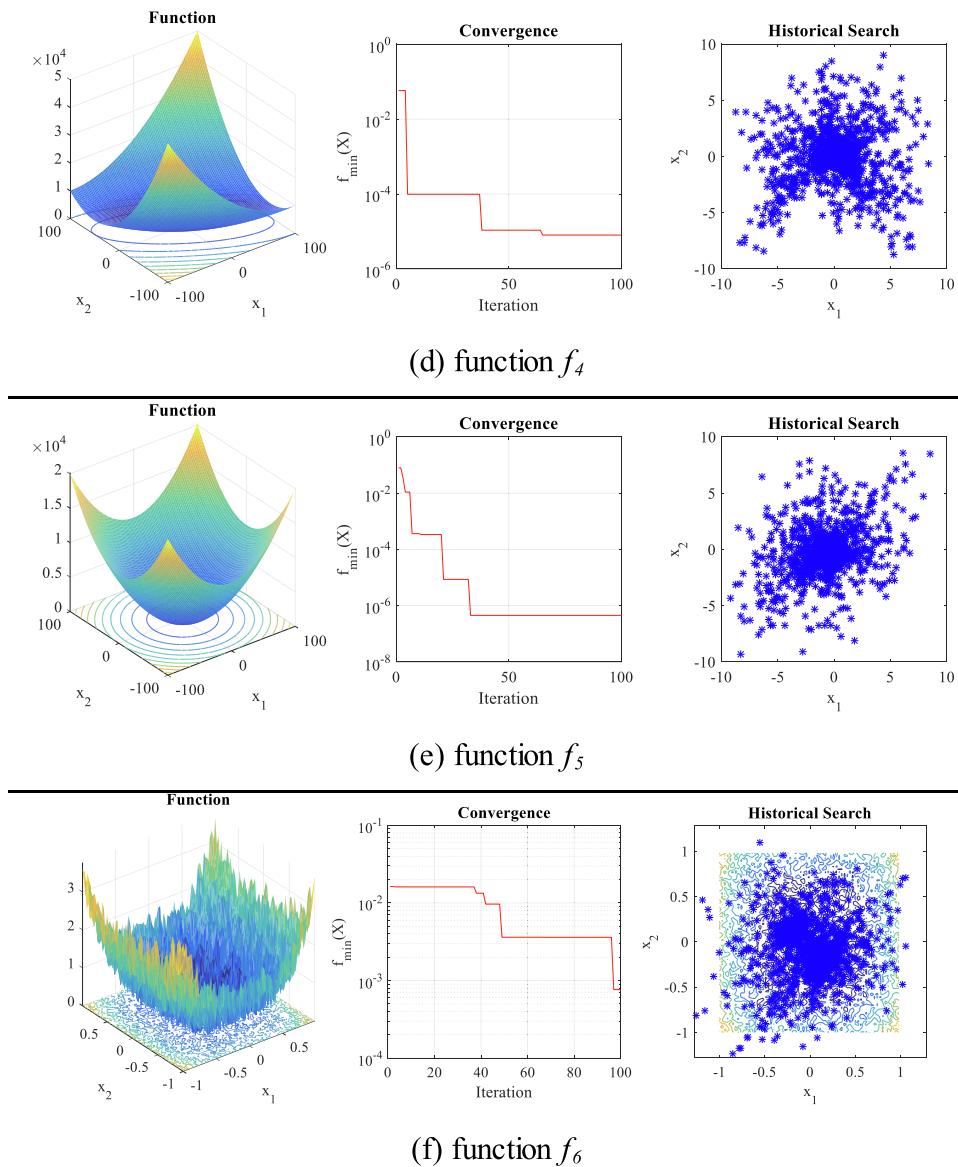


Fig. 11. (continued).

of the LA for all applications of two variables are: $Pop = 20$; $N_{iter} = 100$; $R_c = 150$; $N_p = 10^5$; $S = 1$; $ref = 0.2$ e $M = 0$. These values were defined by the authors after few tries. Pop is the number of points in the cluster that will be evaluated in the objective function in each iteration; N_{iter} is the number of iterations or the number of shots of Lichtenberg figures; M is the figure switching factor which, because it is null, indicates that a single figure will be generated and repeated in all iterations of one program execution; ref it is the refinement coefficient that creates a second figure identical to the main one to improve exploitation, in this case it has 20% the size of it. Finally, N_p , S and R_c are respectively the number of particles, the stickiness coefficient and the radius of creation of the lichtenberg figure to be created by the DLA theory.

The Table 5 presents the main results of the optimization process including the mean and standard deviation (SD) for the design variables and test functions as well as the target value of the functions. f_{min} is the minimum value found for the objective function, evaluated at the minimum points found for the design variables. The closer f_{min} is to F_{opt} , the better the result of the algorithm.

It can be seen that the LA had results very close to the targets, proving to be effective for two-dimensional optimization. Fig. 11

presents the curve of the minimized function, the convergence of f_{min} along the iterations and the search history throughout all iterations for an execution of the algorithm for the results in Table 6.

The functions f_1 and f_2 were tested for a three-dimensional LF and they were simulated 10 times each. The parameters of the LA are: $Pop = 30$; $N_{iter} = 100$; $R_c = 150$; $N_p = 10^6$; $S = 1$; $ref = 0.2$ e $M = 0$. The optimization results are in Table 6 with the mean and SD. As seen, the LA presented satisfactory results for the optimization of problems with three variables. Table 7 shown the results when working with one or >3 variables with f_1 and f_2 , the program was run 10 times for each function. The LA parameters for these simulations are: $Pop = 10^*d$; $N_{iter} = 100$; $R_c = 150$; $N_p = 10^6$; $S = 1$; $ref = 0.4$ e $M = 0$, where “ d “ is the number of system variables

Analyzing the results above, it is observed that the LA was able to find the minimum of different test functions with different numbers of variables.

5.2. Test problem nonlinear and constrained optimization optimization

To solve this problem, Deb (1991), Coello (2000), Coello and Montes

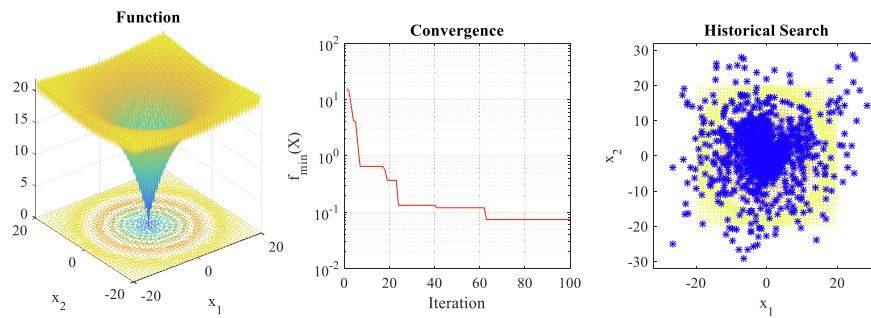
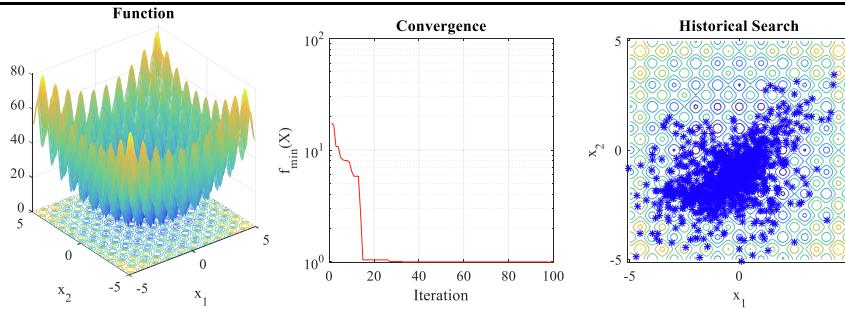
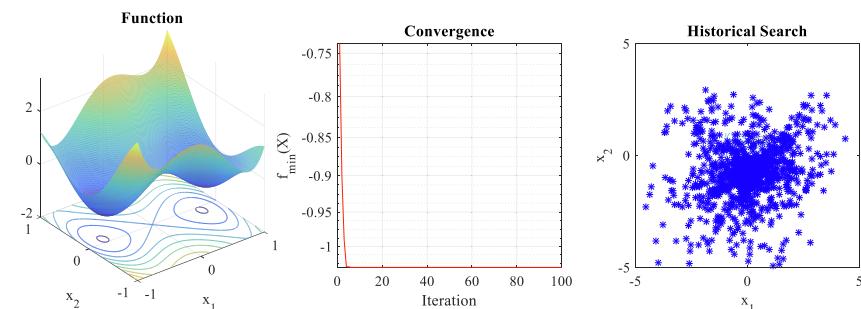
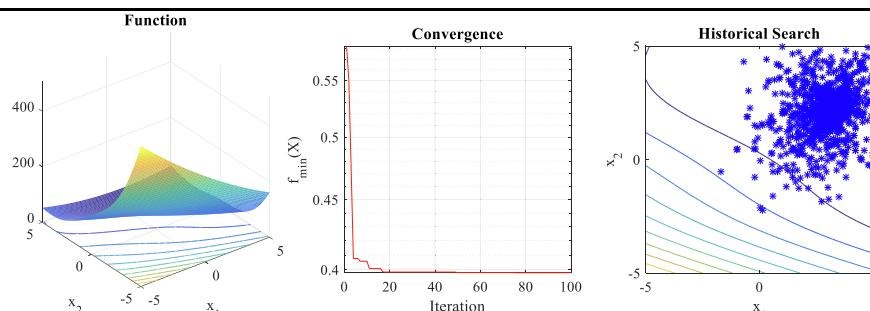
(g) function f_7 (h) function f_8 (i) function f_9 (j) function f_{10} **Fig. 11. (continued).**

Table 6
Validation simulations for three variables.

Run	f_1			f_{min}	F_{opt}	f_2			f_{min}	F_{opt}
	x_1	x_2	x_3			x_1	x_2	x_3		
#1	-0.0020	-0.0025	0.0003	0.0000	0	1.0184	1.0138	1.0293	0.0552	0
#2	0.0007	-0.0005	0.0001	0.0000	0	0.9946	0.9899	0.9806	0.0002	0
#3	0.0101	0.0032	0.0002	0.0001	0	0.9996	1.0002	1.0001	0.0001	0
#4	0.0198	0.0185	0.0453	0.0028	0	0.9991	1.0013	1.0002	0.0017	0
#5	-0.0020	-0.0025	0.0003	0.0000	0	0.9958	0.9896	0.9786	0.0002	0
#6	0.0029	0.0022	0.0025	0.0000	0	0.9670	0.9340	0.8740	0.0059	0
#7	-0.0046	0.0018	0.0014	0.0000	0	1.0062	1.0132	1.0206	0.0039	0
#8	0.0019	0.0013	0.0021	0.0000	0	0.9807	0.9578	0.9187	0.0039	0
#9	0.0013	-0.0005	0.0006	0.0000	0	0.9637	0.9250	0.8514	0.0101	0
#10	0.0007	-0.0005	0.0001	0.0000	0	0.9894	0.9777	0.9592	0.0018	0
Mean	0.0010	0.0004	0.0003	0.0000		0.9952	0.9898	0.9796	0.0039	
SD	0.0071	0.0061	0.0146	0.0009		0.0170	0.0315	0.0609	0.0176	

Table 7
Optimization results for other variables.

Function	Minimum values found		f_{min}	F_{opt}
$f_1 (d = 1)$	Mean	0.0001	0.0002	0
	SD	0.2649	0.0003	0
$f_2 (d = 1)$	Mean	1.0000	0	0
	SD	0.0002	0.0003	0
$f_1 (d = 4)$	Mean	(0.0089, 0.0240, 0.0022, -0.0509)	0.0128	0
	SD	(0.0657, 0.0932, 0.0777, 0.0801)	0.0233	0
$f_2 (d = 4)$	Mean	(0.9888, 0.9828, 0.9685, 1.0133)	0.0110	0
	SD	(0.0270, 0.0525, 0.1025, 0.1839)	0.2683	0
$f_1 (d = 5)$	Mean	(-0.0022, -0.0010, 0.0030, 0.0018, 0.0022)	0.0013	0
	SD	(0.0206, 0.0244, 0.0170, 0.0279, 0.0240)	0.0030	0
$f_2 (d = 5)$	Mean	(1.031, 1.0027, 1.0030, 1.0116, 1.0225)	0.0311	0
	SD	(0.0165, 0.0335, 0.0651, 0.1231, 0.2226)	0.0425	0

(2002) applied Genetic Algorithm; He and Wang (2007) used a version of the Particle Swarm Optimization, the CPSO; Xin-She Yang (2010) used an accelerated version of PSO, the APSO; Kaveh and Talatahari (2010) used Ant Colony Optimization. Finally, Salimi (2014) performed this problem with their Stochastic Fractal Search. The LA parameters are: $Pop = 100$; $N_{iter} = 200$; $R_c = 150$; $N_p = 10^5$; $S = 1$; $ref = 0.2 e M = 0$. Table 8 shows the results of these works as well as the average results after 30 simulations of the LA. All algorithms have the same number of iteration and population. Table 9 shows the average, standard deviation and the best result found in each of these executions.

The welded beam design is extremely complex due to its nonlinear function and its numerous nonlinear constraints. In practical terms, the dimensions obtained by the optimizers are close and consist of an optimal result, the Lichtenberg Algorithm being able to deal with

complex optimization problems. Taking the average of 30 runs as a result, LA had a slightly higher f_{min} result, but managed to have an excellent standard result in one of its runs even ahead of well-consolidated optimizers like the Genetic Algorithm and Accelerated Particle Swarm Optimization.

6. Conclusion

The main objective of this article is to invite the reader to follow in detail the birth of a new optimization algorithm that unites two capabilities found in metaheuristics: an algorithm based on population and at the same time in trajectory.

Ten nonlinear and unconstrained functions were chosen with varying levels of difficulty and the Lichtenberg Algorithm proved to be a powerful optimization tool with excellent results whose differences between the optimal target and the one found had minimal differences with low standard deviations. Lichtenberg Algorithm also faced a nonlinear constrained problem, the Welded Beam Design, where it looked for optimal design points in multimodal and complex regions with limitations beyond that of the search space. He found positive results when compared to those found in the literature, where his best result was better than algorithms established in the literature such as Genetic Algorithm and Particle Swarm Optimization.

With these results, the algorithm that honors Cristoph Lichtenberg has good indications in other optimization problems, such as more complex problems of nonlinear optimization with and without constraints, inverse methods for structural optimization and damage identification through association with a finite element method, optimization of parameters and processes, optimization of sensor locations and even multi-objective optimization.

Table 8
Welded beam design using different algorithms.

Deb (1991)	Coello (2000)	Montes (2002)	He (2007)	Kaveh and Talatahari (2010)	Yang (2010)	Salimi (2014)	LA (present study)
<i>Design Variables</i>							
h	0.2489	0.2088	0.2060	0.2024	0.2057	0.1942	0.2057
l	6.1730	3.4205	3.4713	3.5442	3.4711	3.7409	3.4705
t	8.1789	8.9975	9.0202	9.0482	9.0367	9.0401	9.0366
b	0.2533	0.2100	0.2065	0.2057	0.2057	0.2056	0.2057
<i>Constraints Values (must be negative)</i>							
g_1	-0.0068	-0.3378	-0.1030	-13.655	-0.0846	-	0.0000
g_2	-255.577	-353.903	-0.2317	-75.814	-0.5907	-	-347.5340
g_3	-0.0044	-0.0012	0.0005	-0.0034	0.0000	-	-0.0003
g_4	-2.9829	-3.4119	-3.4300	-3.4246	-3.4329	-	-3.3813
g_5	-0.1239	-0.0838	-0.0810	-0.0774	-0.0807	-	-0.0963
g_6	-0.2342	-0.2356	-0.2355	-0.2356	-0.2355	-	-0.2353
g_7	-0.0045	-363.232	-58.646	-4.4729	-0.1448	-	0.0000
f_{min}	2.4332	1.7483	1.7282	1.7280	1.7280	1.7514	1.7249
							1.8446

Table 9

Statistical results of the objective function for the welded beam design for different algorithms.

Result	Deb (1991)	Coello (2000)	Montes (2002)	He (2007)	Kaveh & Talatahari (2010)	Yang (2010)	Salimi (2014)	LA (present study)
Best	2.4331	1.7483	1.7282	1.7280	1.7249	1.7249	1.7465	1.7351
Mean	–	1.7720	1.7927	1.7488	1.7298	1.7249	1.7514	1.8446
Std	–	0.0112	0.0747	0.0129	0.0092	0.0000	0.1171	0.1597

CRediT authorship contribution statement

João Luiz Junho Pereira: Conceptualization, Methodology, Software, Formal analysis, Investigation, Writing - original draft, Writing - review & editing. **Matheus Brendon Francisco:** Formal analysis, Investigation, Writing - review & editing. **Camila Aparecida Diniz:** Formal analysis, Investigation, Writing - review & editing. **Guilherme Antônio Oliver:** Formal analysis, Investigation. **Sebastião Simões Cunha:** Formal analysis, Investigation, Writing - review & editing. **Guilherme Ferreira Gomes:** Conceptualization, Methodology, Software, Writing - original draft, Writing - review & editing, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors would like to acknowledge the financial support from the Brazilian agency CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico), CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) and FAPEMIG (Fundação de Amparo à Pesquisa do Estado de Minas Gerais - APQ-00385-18). The authors would also like to acknowledge of the Tutorial Education Program (PET – Programa de Educação Tutorial).

References

- Deb, K. (1991). Optimal design of a welded beam via genetic algorithms. *AIAA Journal*, 29(11), 2013–2015. <https://doi.org/10.2514/3.10834>
- Coello Coello, C. A. (2000). Use of a self-adaptive penalty approach for engineering optimization problems. *Computers in Industry*, 41(2), 113–127. [https://doi.org/10.1016/S0166-3615\(99\)00046-9](https://doi.org/10.1016/S0166-3615(99)00046-9)
- Coello Coello, C. A., & Mezura Montes, E. (2002). Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Advanced Engineering Informatics*, 16(3), 193–203. [https://doi.org/10.1016/S1474-0346\(02\)00011-3](https://doi.org/10.1016/S1474-0346(02)00011-3)
- Chopard, B., & Tomassini, M. (2018). *Problems, Algorithms, and Computational Complexity: An Introduction to Metaheuristics for Optimization* (pp. 1–14). Springer.
- Di Rocco, E. D., D. L. A., & Model. (2009). *Seminar Univerza V Ljubljani*.
- Dorigo, M. (1992). Optimization, learning and natural algorithms. *Ph.D. thesis, Politecnico di Milano, Italy*.
- Elaziz, M. A., & Mirjalili, S. (2019). A hyper-heuristic for improving the initial population of whale optimization algorithm. *Knowledge-Based Systems*, 172, 42–63. <https://doi.org/10.1016/j.knosys.2019.02.010>
- Geem, Z. W., Hoon Kim, J., & Loganathan, G. V. (2001). A New Heuristic Optimization Algorithm: Harmony Search. *SIMULATION*, 76(2), 60–68.
- Gomes, G. F., da Cunha, S. S., Jr., & Ancelotti, A. C., Jr. (2019). A sunflower optimization (SFO) algorithm applied to damage identification on laminated composite plates. *Engineering with Computers*, 35(2), 619–626. <https://doi.org/10.1007/s00366-018-0620-8>
- Gomes, G. F., & de Almeida, F. A. (2020). Tuning metaheuristic algorithms using mixture design: Application of sunflower optimization for structural damage identification. *Advances in Engineering Software*, 149, 102877. <https://doi.org/10.1016/j.advengsoft.2020.102877>
- Gomes, G. F., & Giovani, R. S. (2020). An efficient two-step damage identification method using sunflower optimization algorithm and mode shape curvature (MSDBI-SFO). *Engineering with Computers*. <https://doi.org/10.1007/s00366-020-01128-2>
- Gomes, G. F., & Pereira, J. V. P. (2020). Sensor placement optimization and damage identification in a fuselage structure using inverse modal problem and firefly algorithm. *Evolutionary Intelligence*, 13(4), 571–591. <https://doi.org/10.1007/s12065-020-00372-1>
- Hatamlou, A. (2013). Black hole: A new heuristic optimization approach for data clustering. *Information sciences*, 222, 175–184.
- Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems*, 97, 849–872. <https://doi.org/10.1016/j.future.2019.02.028>
- Holland, J. (1975). *Adaptation in natural and artificial systems*. Ann Arbor, MI, USA: University of Michigan Press.
- Irurzun, I. M., Bergero, P., Mola, V., Cordero, M. C., Vicente, J. L., & Mola, E. E. (2002). Dielectric breakdown in solids modeled by BDM and DLA. Chaos, Solitons and Fractals.
- Kaveh, A., & Talatahari, S. (2010). An improved ant colony optimization for constrained engineering design problems. *Engineering Computations*, 27(1), 155–182. <https://doi.org/10.1108/02644401011008577>
- Kennedy, J., Eberhart, R. C. (1995). Particle swarm optimization. In Proceedings of the IEEE international conference on neural networks (pp. 1942–1948).
- Lichtenberg, G. C. (1777). Novi. Comment. Gött. (Vol. 8, p. 168).
- Meakin, P., Tolman, S., & Blumen, A. (1989). Diffusion-Limited Aggregation. The Royal Society.
- Merrill, F. H., & Von Hippel, A. (1939). The atom physical interpretation os Lichtenberg figures and their application to the study of gas discharge. *Journal of Applied Physics*, 10(12), 873–887.
- Mirjalili, S., & Lewis, A. (2016). Moth-flame optimization akgorithm: a novel nature-inspired heuristic paradigm. *Knowledge-Based Systems* 89, November 2015.
- Mirjalili, S. (2015). The whale optimization algorithm. *Advances in Engineering Software*, 95, 51–67.
- Naccarato, K. P. (2001). Estudo de relâmpagos no Brasil com base na análise de desempenho do sistema de localização de tempestades / K. P. Naccarato – São José dos Campos: INPE, 165p. – (INPE-8380-TDI/770).
- Nabil, E. (2016). A modified flower pollination algorithm for global optimization. *Expert Systems with Applications*, 57, 192–203. <https://doi.org/10.1016/j.eswa.2016.03.047>
- Niemeyer, L., Pietronero, L., & Wiesmann, H. J. (1984). *Phys Ver Lett*, 52, 1033.
- Olorunda, O., Engelbrecht, A. P. (2008). Measuring exploration/exploitation in particle swarms using swarm diversity. In IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence) (pp. 1128–1134). IEEE.
- Patel, V. K., & Savsani, V. J. (2015). Heat transfer search (HTS): A novel optimization algorithm. *Information Sciences*, 324, 217–246. <https://doi.org/10.1016/j.ins.2015.06.044>
- Pereira, J. L. J., Chuman, M., Cunha, S. S. Jr., & Gomes, G. F. (2020). Lichtenberg optimization algorithm applied to crack tip identification in thin plate-like structures. *Engineering & Computations* <https://doi.org/10.1108/EC-12-2019-0564>.
- Pinto, I. R. C. A. (1987). Estudos sobre campos elétricos e condutividade associados a nuvens eletrificadas na região da América do Sul. São José dos Campos. 129p. (INPE-4487-TDL/325). Dissertation (Doctorate in Space Geophysics) - Instituto Nacional de Pesquisas Espaciais.
- Askari, Q., Younas, I., & Saeed, M. (2020). Political Optimizer: A novel socio-inspired meta-heuristic for global optimization. *Knowledge-Based Systems*, 195, 105709. <https://doi.org/10.1016/j.knosys.2020.105709>
- He, Q., & Wang, L. (2007). An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Engineering Applications of Artificial Intelligence*, 20(1), 89–99. <https://doi.org/10.1016/j.engappai.2006.03.003>
- Rao, S. S., & Rao, S. S. (2009). Engineering optimization: theory and practice. John Wiley & Sons.
- Rashedi, E., Nezamabadi-pour, H., & Saryazdi, S. (2009). GSA: a gravitational search algorithm. *Information Sciences*, 179(13), 2232–2248. <https://doi.org/10.1016/j.ins.2009.03.004>
- Salimi, H. (2014). Stochastic Fractal Search: A powerful metaheuristic algorithm. *Knowledge Based Systems*. <https://doi.org/10.1016/j.knosys.2014.07.025>
- Shareef, H., Ibrahim, A. A., & Mutlag, A. H. (2015). Lightning search algorithm. *Applied Soft Computing*, 36, 315–333. <https://doi.org/10.1016/j.asoc.2015.07.028>
- Storn, R., & Price, K. (1997). Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces. *J Global Optimization*, 11, 341–359.
- Turner, A. (2019). From Lichtenberg to lightning: Understanding random growth. *Newsletter of the London Mathematical Society*.
- Witten, T. A., & Sander, L. M. (1981). Diffusion-limited aggregation, a kinetic critical phenomenon. *Physical Review Letters*, 47(19), 1400–1403. <https://doi.org/10.1103/PhysRevLett.47.1400>
- Witten, T. A., & Sander, L. M. (1983). Diffusion-limited aggregation. *Physical Review B*, 27 (9), 5686–5697. <https://doi.org/10.1103/PhysRevB.27.5686>
- Xu, Y., Cui, Z., & Zeng, J. (2010). Social Emotional Optimization Algorithm for Nonlinear Constrained Optimization Problems. *Lecture Notes in Computer Science*, 583–590.
- Yang, X.-S. (2010). *Nature-inspired Metaheuristic Algorithms* (Second Edition). Luniver Press.
- Yang, X.-S. (2014). *Nature-inspired optimization algorithms*. Elsevier.