

Article

A novel geo-inspired earthquake optimization algorithm

Efrain Mendez ^{1,†,‡}, Alexandro Ortiz ^{1,‡*}, Pedro Ponce ^{2,‡} and Arturo Molina ^{1,‡}

¹ Tecnológico de Monterrey, School of Engineering and Science, Mexico; a01336094@itesm.mx

² Tecnológico de Monterrey, School of Engineering and Science, Mexico; a00787745@itesm.mx

³ Tecnológico de Monterrey, School of Engineering and Science, Mexico; pedro.ponce@itesm.mx

⁴ Tecnológico de Monterrey, School of Engineering and Science, Mexico; armolina@itesm.mx

* Correspondence: a00787745@itesm.mx

‡ These authors contributed equally to this work.

Academic Editor: name

Version September 11, 2018 submitted to

Abstract: A novel metaheuristic optimization method is proposed based on an earthquake that is a geology phenomenon. The novel Earthquake Algorithm (EA) proposed, adapts the principle of propagation of geology waves P and S through the earth material composed by random density to ensure the dynamic balance between exploration and exploitation, in order to reach the best solution to optimization complex problems by searching for the optimum into the search space. The performance and validation of the EA are compared against the Bat Algorithm (BA) and the Particle Swarm Optimization (PSO) by using 10 diverse benchmark functions. In addition, an experimental engineering application is implemented to evaluate the proposed algorithm. Early results show a feasibility of the proposed method with a clearly constancy and stability. It is important highlight the fact that the main purpose of this paper is to present a new line of research, which is opened from the novel EA.

Keywords: Optimization; Metaheuristic; Earthquake Algorithm; Bat Algorithm; Particle Swarm Optimization; PID controller; DC motor; Fuzzy Logic; Mamdani; Geo-inspired computing.

1. Introduction

Optimization techniques are used to solve problems choosing the best option between all possible solutions, the metaheuristic algorithms are becoming powerful methods tough optimization problems [1].

According to [2], optimization algorithms can be divided into deterministic and stochastic classes. Deterministic methods are classified into evaluation and gradient-based methods. Evaluation methods, where calculating gradient functions is not necessary but they are extremely slow and ineffective. Gradient-based methods utilize gradients or derivations of the objective function to direct the exploration. Nevertheless, there is not security that the convergence to the global optimum by these methods is realized in objective functions with rough and complex forms. Otherwise, stochastic methods are not required gradient information to try to find the global optimum but they need to execute the objective function in order to find the optimum point [3].

In real and complex design problems, optimization algorithms may fail or result a local optima. Such problems cannot be resolved by conventional methods that locate local optima when the objective is to find the global optimum [4].

Metaheuristic algorithms offer an effective solution strategies due their distinct mechanisms to avoid from locally optimal solutions [5]. The randomization procedure creates unsystematic solutions, which objective is to find a global solution through search space. By the local search, convergence and focus on realize good solutions in a region is determined [6].

In many instances, metaheuristics optimization algorithms have the same characteristics: (1) nature-inspired (with principles from physics, biology, geography, ethology), (2) stochastic components

is used (including random variables), (3) gradient or Hessian matrix of the objective function is not used and (4) some parameters need to be adjusted on line with the problem to resolve [7]. Another important characteristic is the strategy for exploring search spaces. A dynamic balance from diversification and intensification must be given. The first term allude to the examination of the search space, whereas the second term refers to exploitation according to search knowledge [8].

In four principal categories, metaheuristics algorithms are classified: (1) evolutionary-based, (2) physical-based, (3) swarm-based and (4) human behavior [9]. Evolutionary-based algorithms imitate the natural biological evolution and/or social behavior of species [10]. Some techniques of evolutionary-based are: Genetic Algorithms (GA) [11], Evolutionary Programming (EP) [12], Evolutionary Strategy (ES) [13], Differential Evolution (DE) [14].

In the second category, physical-based algorithms have some characteristics in rules of physics such as gravitational, electromagnetic and inertia force, etc. These rules govern the behavior of the communication or movement through of search agents [15]. Popular algorithms of physical-based are: Simulated Annealing (SA) [16], Gravitational Search Algorithm (GSA) [17], Magnetic Optimization Algorithm (MOA) [18], Charged System Search (CSS) [19], Central Force Optimization (CFO), Black Hole (BH) [20], Galaxy-based Search Algorithm (GbSA) [21], Artificial Chemical Reaction Algorithm (ACROA) [22], Small-World Optimization Algorithm (SWOA) [23].

The third category, swarm-based algorithms use the collective intelligence from organisms in a community sharing local information with other individuals, such as ants and bees [24]. The most known methods are: Particle Swarm Optimization (PSO) [25], Monkey Search Algorithm (MSA) [26], Bat Algorithm (BA) [27], Cuckoo Search (CS) [28], Wolf pack search algorithm [29], Firefly Algorithm (FA) [30], Artificial Bee Colony (ABC) [31], Ant Colony Optimization (ACO) [32].

The last category is based from human behavior in different situations or activities such as: education behavior (Teaching-learning-based Optimization, TLBO) [33], neighborhood searching (Tabu-search) [34], socio-politically motivated strategy (Imperialist Competitive Algorithm, ICA) [35], musicians looking for the best melody (Harmony Search, HS) [36], mimic sport teams competition (League Championship Algorithm, LCA) [37]. Fig. 1 depicts the broadly classification from metaheuristic optimization algorithms.

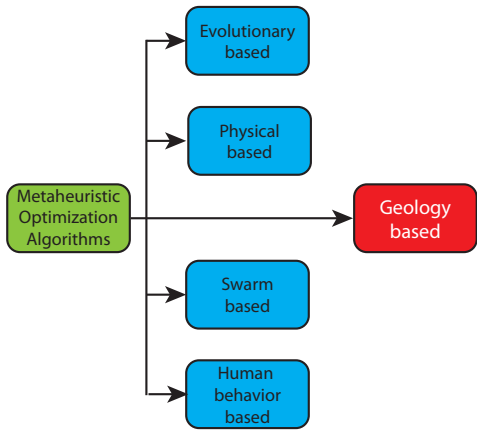


Figure 1. Classification of metaheuristic optimization algorithm.

Nevertheless, it is important highlight the fact that the main purpose of this paper is to present a new line of research, which is opened from the novel algorithm presented throughout this document.

In order to achieve the objective, Section 2 explains the basic concepts of an Earthquake behavior, then Section 3 shows the algorithm inspired by it. Later, in order to prove the viability of the proposed algorithm Section 4 shows its performance against some commonly used benchmark functions, and Section 5 its behavior against a real implementation (Speed controller optimization of a DC motor). Finally, Section 6 opens the discussion of the obtained results and Section 7 explains the conclusions.

2. Earthquake background

Earthquakes are located in the “top list” of the most catastrophic and destructive natural disasters provoking fatalities in human life and economic lost [38]. In seismic zones, elastic energy collects around those regions in advance of friction is outdo and unexpected shear movements occurs in form as an earthquake, decreasing surrounding strain energy and generating seismic waves that shake the external layer of the Earth [39].

From [40], seismological community has been discussing in the last decade that the first few seconds of the P-wave provides information about the final magnitude of an earthquake. This allows to use P-wave velocity to make a good exploration by delivering information from search space to find the optimum solution.

In [41], mentioned that seismic waves are elastic waves and each earth material must behave this characteristic to transmit them. Elasticity degree dictates the transmission capacity. By an earthquake, earth material is submitted to stress (compression, tension and/or shearing). Earth material has a ductile characteristic at very slow strain rates, as movements in the order of mm or cm/year. Conversely, the earth acts elastically to fast and small deformations caused by earthquakes. When high amplitude of seismic deformations and long-period free-oscillations modes are present, the inelastic response of seismic waves must be considered.

According to [42], earth material behavior can be defined by Hooke’s Law [43] in elastic range. Above its elastic limit, earth material can be respond with brittle fracturing (e.g. fault by earthquake) or ductility.

Depending on the type of deformation, elastic material endures to stress and can be quantified by some elastic moduli [41]:

- *bulk modulus* κ is described as the ratio of the hydrostatic pressure change to the resulting relative volume change (i.e. $\kappa = \Delta P / (\Delta V / V)$).
- *Lamé parameter* μ (*shear modulus*) is the resistance of the material to shearing, namely, to changing the shape and not volume (i.e. $\mu = \tau_{xy} / 2e_{xy}$ or $\mu = (\Delta F / A) / (\Delta L / L)$).
- *Young’s modulus* E is defined by the response of a cylinder with Length which is pulled on both ends. Its value is given by $E = (F / A) / (\Delta L / L)$.
- *Poisson’s ratio* σ is the ratio between cylinder lateral contraction being pulled on its ends to its relative longitudinal extension, i.e. $\sigma = (\Delta W / W) / (\Delta L / L)$.
- *Lamé parameter* λ does not have explanation physical but it can be defined in terms of the elastic moduli already mentioned $\lambda = \sigma E / ((1 + \sigma)(1 - 2\sigma))$
- *density* ρ of earth material. Denser rocks have faster wave propagation because rigidity increases with density.

2.1. P and S waves velocities

When a earthquake occurs, rocks break and generate waves through interior to surface of the Earth, Fig. 2. There are two types of waves P and S that are generated. The P -wave is the fastest and depends on the earth material compressibility. They are transmitted by compression and tension of the medium with volume changes. The S -wave is slower than P -wave, depends on rocks elasticity and move the epicenters up and down, perpendicular to the wave direction.

Eqs. (1) and (2) describe the estimation of P and S waves velocities.

$$v_p = \sqrt{\frac{\lambda + 2\mu}{\rho}} \quad (1)$$

$$v_s = \sqrt{\frac{\mu}{\rho}} \quad (2)$$

where v_p and v_s are the P and S waves speed, λ and μ the Lamé parameters, and ρ the density of earth material.

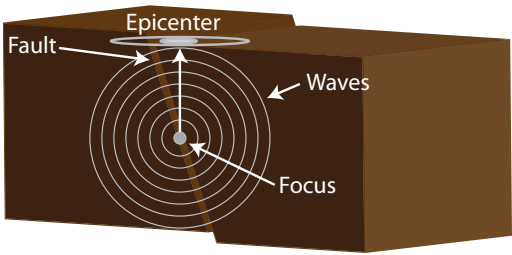


Figure 2. Earthquake origins.

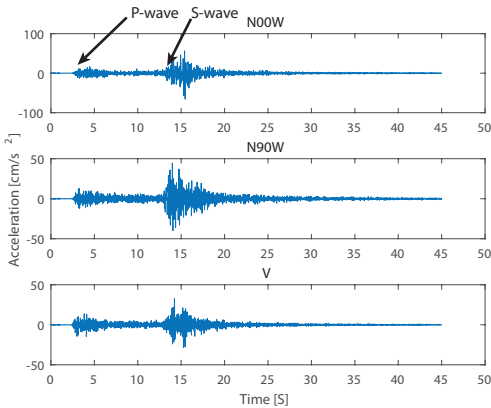


Figure 3. Ground acceleration (3-axis) from an earthquake record.

113 Records from ground-acceleration (Fig. 3) shows the waves *P* and *S* when an earthquake occurred
114 in Chiapas, Mexico (09.15.2010) with an epicenter in 15.59 N, 93.52 W at 95 km of profundity [44].
115 *P*-waves are the first to appear (due to faster velocity) and are detected by seismograph accelerometers.
116 Seconds after, *S*-waves are observed.

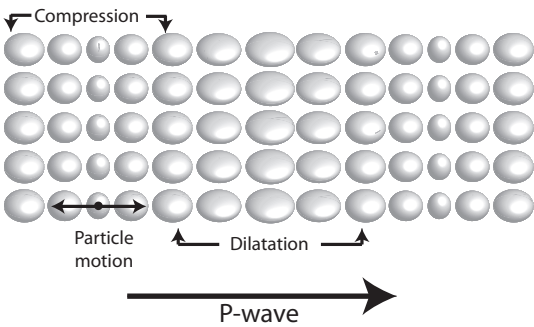


Figure 4. Travel of *P*-wave through an earth material by compression and dilation.

117 In the Fig. 4 can be visualized the movement of the *P*-waves through a medium by compression
118 and dilation with volume changes. These kind of waves can be transmitted by any medium (solid,
119 liquid and gas). Particle motion of earth material are represented by spheres.

120 Representation of the *S*-waves movement is shown in the Fig. 5, the propagation only occurs in
121 solid medium with shearing deformation (perpendicular movements to the wave direction).

122 **3. Earthquake optimization algorithm**

123 The Earthquake algorithm initially works with a random population of solutions, which are called
124 epicenters. Besides, it also uses the implementation of a function for the random motion in order to

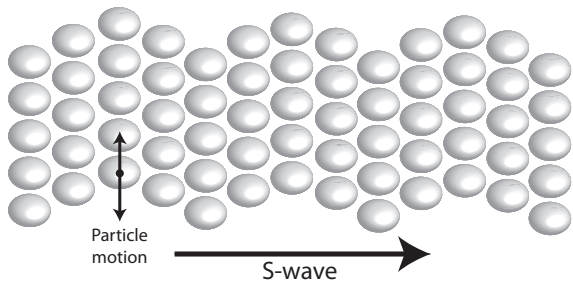


Figure 5. Travel of S-wave through an earth material by shearing deformation.

125 produce some members of the epicenters in distant regions of the searching space. The first version of
126 this Earthquake algorithm was proposed in [45].

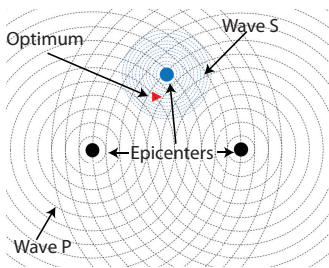


Figure 6. Population of three epicenters.

127 Then, as will be explained later in this section, some of the parameters used for this novel
128 algorithm can be described using a *Poisson ratio*, reason that inspired the implementation of an
129 *Exponential Distribution*, from the relation between a Poisson and an Exponential distribution taken
130 from [46], for the random generation already mentioned.
131 As seen in Section 2, the motion of an earthquake can be parametrized with the velocity equations
132 of the *P* and *S* waves, where the principle of operation of the proposed algorithm lies in those
133 parameters. Knowing then, that the transmission of the *P*-wave is faster than that of the *S*-wave, the
134 *P*-wave is used for a quick exploration and the other one for a more detailed one.

Table 1. Principle materials or geologic formations properties [41].

Material or geologic formation	Density (Kg/m^3)	Poisson ratio	Vp/Vs
Sandstone	2500	0.21	1.65
Salt	2200	0.17	1.59
Limestone	2700	0.19	1.62
Granite	2610	0.25	1.73 (1.65-1.91)
Basalt	2940	0.28	1.8 (1.76-1.82)
Peridotite, Dunit, Pyroxenite	3300	0.29	1.8 (1.76-1.91)

135 In order to determine when to use a wave or the other, it is essential for the algorithm to define an
136 operation range for the *S*-wave, which will be referred to in this document as the *S-range* or *Sr*. Also,
137 as seen in Fig. 7, the *Sr* is defined around the best solution.

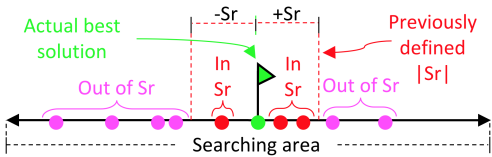


Figure 7. Procedure to find particles in and out the defined *Sr*.

The S_r , should be assigned with the previous knowledge of the problem needs, nevertheless it is recommended to implement the range in function of the percentage of error, between the obtained solutions and the expected ones.

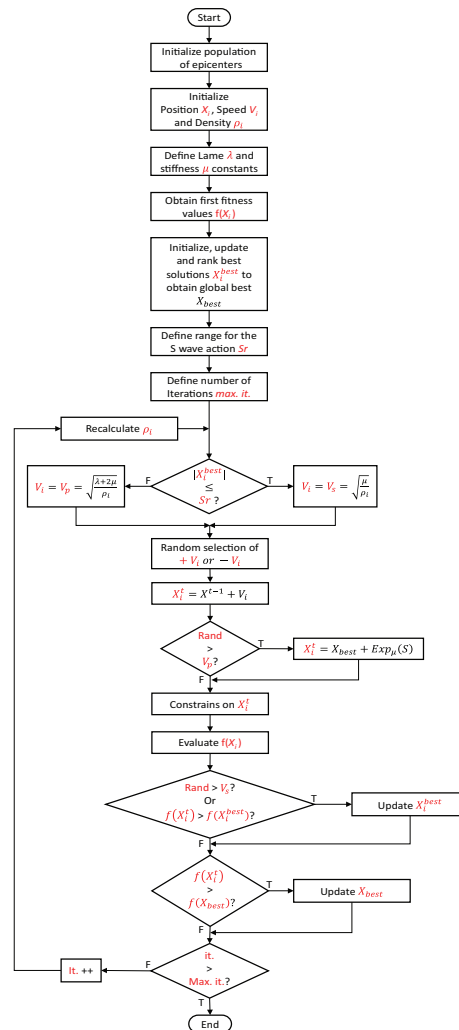


Figure 8. Proposed architecture for the Earthquake Algorithm.

Thus, in Fig. 6 can be seen a population of three epicenters, where the black epicenters are out of S_r , reason why they use the velocity of P -waves (black dotted lines), to map where the solution is (red triangle). The blue epicenter is located in smaller distance from the objective, it is also in S_r and that is why it uses the S -wave (blue dotted lines) for the mapping.

Knowing that the velocity of the P and S waves are given by Eqs. (1) and (2) respectively, it is also known that the Lamé parameters (λ and μ) and the density (ρ), are needed in order to determine the current transmission speed.

To contextualize property values analyzed in this work, Table 1 shows the principle values of density, Poisson's ratio and ratio of seismic wave velocities from some earth materials (average and/or ranges).

Now, in order to validate the proposed constant for the Lamé parameters, [41] explains that most of the rocks have a *Poisson ratio* between 0.2 and 0.3, leaving a mean optimal value on 0.25 for the ratio. As well, also taken from [41], the following equation shows the relation between the Poisson ratio and the Lamé parameters:

```

1: Define objective function.
2: Initialize population of epicenters.
3: Randomly initialize  $x_i$ .
4: Initialize  $v_i$  and  $\rho_i$ .
5: Define  $\mu$  and  $\lambda$ .
6: Obtain and rank the first fitness values.
7: Initialize  $x_{best}$  and  $x_{best}$  with the fitness values.
8: Define  $Sr$ , and the  $Max. it.$ 
9: while ( $it. < Max. it.$ ) do
10:   Randomly calculate  $\rho_i$ .
11:    $v_p \leftarrow \sqrt{\frac{\mu}{\rho_i + 2\mu}}$ 
12:    $v_s \leftarrow \sqrt{\frac{\rho_i}{\lambda + 2\mu}}$ 
13:   if ( $|x_{best} - \rho_i| \leq Sr$ ) then
14:      $v_i \leftarrow v_s$ 
15:   else
16:      $v_i \leftarrow v_p$ 
17:   end if
18:   Randomly select positive or negative  $v_i$ .
19:    $x_i^t \leftarrow x_i^{t-1} + v_i$ 
20:   if ( $rand > v_p$ ) then
21:      $x_i^t \leftarrow x_{best} + Exp_{\mu}(s)$ 
22:   end if
23:   Constrain  $x_i$  if needed.
24:   Obtain the new fitness values.
25:   Actualize  $x_{best}$  and  $x_i^{best}$ 
26: end while
27: Postprocessing and visualization of results.

```

Figure 9. Pseudocode of the Earthquake Algorithm

$$\sigma = \frac{\lambda}{2(\lambda + \mu)} \quad (3)$$

where σ is the Poisson's ratio.

According to [41], the Lamé parameters can be the same under some circumstances, so for the current algorithm it is taken that $\lambda = \mu$. In that case, in order to find the optimal Lamé parameters to be used, several tests were performed with different Lamé values, finding that the only real constant that worked was 1.5; that is the reason why, for the algorithm purpose, the wave transmission parameters can be defined as the constant 1.5, taken from Table 1, giving to:

$$\lambda = \mu = 1.5 \text{ GPa} \quad (4)$$

Where substituting Eq. (4) on Eq. (3):

$$\sigma = \frac{1.5}{2(1.5 + 1.5)} = 0.25 \quad (5)$$

Which validates the premise developed to obtain Eq. (4). On the other hand, for the implementation of the algorithm, the density of the solids (ρ) is used as a random value, selected from a range between 2200 and 3300 Kg/m³, also according to Table 1.

Shown the initial position of the epicenters population are randomly selected, however their speed are initially 0. On the other hand, the update of the current position of the epicenters, is given by Eq. (6):

$$X_i^t = X_i^{t-1} + V_i \quad (6)$$

where X_i^t and X_i^{t-1} are the current and the previous positions, meanwhile the V_i is the current speed.

Finally, the equation that incorporates the Exponential distribution, to reduce the probability of visiting points already visited for the epicenters, or epicenters is described in [46]. Where the random value is generated with the distribution, in a range of \pm the maximum value of V_p/V_s , that is ± 1.91 taken from Table 1.

$$X_i^t = X_{best} + Exp_{\mu}(s) \quad (7)$$

where X_{best} is the global best solution, and the $Exp_{\mu}(s)$ is the random value generated with the exponential distribution from the value of μ .

Whence, the random update for the position with the Exponential distribution for this algorithm, is given by Eq. (7), and the diagram that describes the proposed architecture for the earthquake algorithm is shown by Fig. 8.

Hence, the Sr is recommended to be $\pm 10\%$ from the best solution, but after selecting if the current epicenter is going to use the v_s or v_p as v_i (see Fig. 8), it results very important for the algorithms performance to understand that since both speeds are calculated by a square root, the final result is a positive number, but it is also known that the result contemplates a $\pm v_i$.

The above is the reason why the flowchart, that describes the proposed architecture of the algorithm, contemplates the use of a random selection of a positive or a negative v_i , just to give to the EA another degree of freedom.

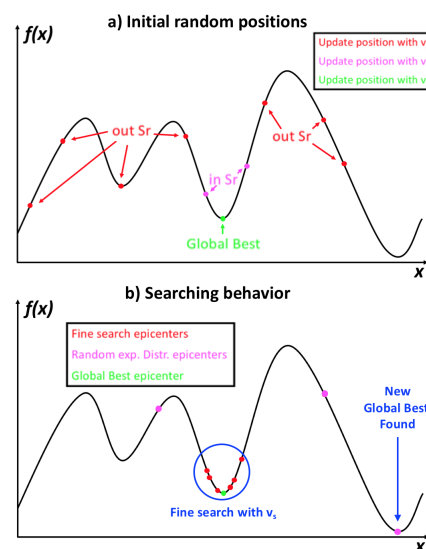


Figure 10. Behavior of the Earthquake algorithm using ten epicenters.

To clarify how the proposed algorithm works, Fig. 9 shows the pseudocode of the Earthquake Algorithm. Meanwhile Fig. 10 shows the generalities of the Earthquake Algorithm behavior, which in Fig. 10(a) appears an example of how could the epicenters randomly be placed on a function, same as when evaluated give the first fitness values. After the first rank of the best solutions, it can be known which of the epicenters are in or out the Sr and hence, which of them are going to use the v_s or the v_p speeds to update their positions.

Then, in Fig. 10(b), the searching behavior is shown, where it can be seen that after some iterations the epicenters start to converge, with a fine search around the current global best. In spite of this, it can be observed that three epicenters took distant routes from the epicenters set, because of the random generation of positions using the exponential distribution (previously explained).

As already said, that final generation of some epicenters allowed the algorithm to “escape” the local minimum, what caused that another epicenter found a better solution. Replacing the previous global best solution, the epicenters sooner or later would converge again to begin another fine search, keeping the possibility that another randomly generated epicenter finds another “best solution”.

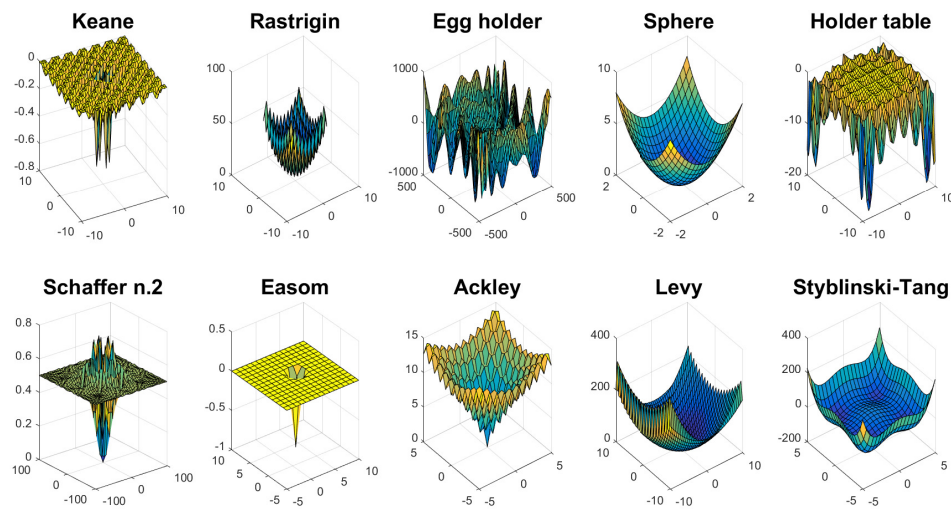


Figure 11. Surfaces of the Benchmark functions used to compare the performance of the EA against the PSO and BA [47,48].

Table 2. Parameters obtained for plant models using each optimization algorithm (EA, BA and PSO)

Function	Formula	Global minimum	Search domain
Keane	$f(x, y) = -\frac{\sin^2(x-y)\sin^2(x+y)}{\sqrt{x^2+y^2}}$	$\text{Min} = \begin{cases} f(1.393, 0) = 0.674 \\ f(0, 1.393) = 0.674 \end{cases}$	$-10 \leq x, y \leq 10$
Rastrigin	$f(x) = 10d + \sum_{i=1}^d [x_i^2 - 10\cos(2\pi x_i)]$ where $d = 2$	$f(0, 0) = 0$	$-5.12 \leq x_i \leq 5.12$
Egg holder	$f(x, y) = -(y + 47)\sin\sqrt{ \frac{x}{2} + (y + 47) } - x\sin\sqrt{ x - (y + 47) }$	$f(512, 404.23) = -959.64$	$-512 \leq x, y \leq 512$
Sphere	$f(x) = \sum_{i=1}^d x_i^2$	$f(0, 0) = 0$	$-\infty \leq x_i \leq \infty$
Holder table	$f(x, y) = - \sin(x)\cos(y)\exp(1 - \frac{\sqrt{x^2+y^2}}{\pi}) $	$\text{Min} = \begin{cases} f(8.05, 9.66) = -19.21 \\ f(-8.05, 9.66) = -19.21 \\ f(8.05, -9.66) = -19.21 \\ f(-8.05, -9.66) = -19.21 \end{cases}$	$-10 \leq x, y \leq 10$
Schaffer N. 2	$f(x, y) = 0.5 + \frac{\sin^2(x^2 - y^2) - 0.5}{[1 + 0.001(x^2 + y^2)]^2}$	$f(0, 0) = 0$	$-100 \leq x, y \leq 100$
Easom	$f(x, y) = -\cos(x)\cos(y)\exp(-((x - \pi)^2 + (y - \pi)^2))$	$f(\pi, \pi) = -1$	$-100 \leq x, y \leq 100$
Ackley	$f(x, y) = -20\exp[-0.2\sqrt{0.5(x^2 + y^2)}] - \exp[0.5(\cos 2\pi x + \cos 2\pi y)] + e + 20$	$f(0, 0) = 0$	$-5 \leq x, y \leq 5$
Levy N. 13	$f(x, y) = \sin^2 3\pi x + (x - 1)^2(1 + \sin^2 3\pi y) + (y - 1)^2(1 + \sin^2 2\pi y)$	$f(1, 1) = 0$	$-10 \leq x, y \leq 10$
Styblinski-Tang	$f(x) = \frac{1}{2} \sum_{i=1}^d (x_i^4 - 16x_i^2 + 5x_i)$ where $d = 2$	$f(-2.9, -2.9) = -39.166d$	$-5 \leq x_i \leq 5$

4. Benchmark functions

To validate the proposed optimization method mentioned in Section 3, the performance of the Earthquake Algorithm is tested in this section by 10 benchmark functions, and its performance compared against the Particle Swarm Optimization and the Bat algorithm.

Standard test functions are used to evaluate optimization algorithms, the benchmark functions selected were as diverse as possible to analyze how the algorithm performs between different conditions, in a convergence problem as the convex *Spheric function*, against unimodal and non-convex functions like the *Ackley function*, or even under multimodal non-convex functions with several local or global solutions, like the *Egg holder function* or the *Holder table function*.

Table 3. Optimization algorithms performances from different benchmark functions.

Function	AVG			STD			BEST		
	EA	BA	PSO	EA	BA	PSO	EA	BA	PSO
Keane	-0.672	-0.591	-0.672	0.002	0.155	0.002	-0.674	-0.674	-0.674
Rastrigin	0.758	2.117	3.601	0.454	3.136	5.349	0.007	1.0e ⁻⁵	1.2e⁻⁵
Egg holder	-744.332	-838.671	-770.183	133.461	75.523	72.209	-956.917	-956.290	-931.974
Sphere	0.005	0.028	2.8e⁻⁶	0.005	0.061	4.7e⁻⁶	9.5e ⁻⁵	9.3e ⁻⁹	9.1e⁻¹²
Holder table	-19.153	-18.324	-17.593	0.057	1.434	0.992	-19.209	-19.209	-19.194
Schaffer N. 2	1.8e ⁻⁴	0.084	7.8e⁻⁹	0.002	0.130	1.9e⁻⁸	2.6e ⁻⁸	1.9e ⁻⁷	2.4e⁻¹³
Easom	-0.772	-0.353	-0.793	0.412	0.393	0.175	-0.999	-1.000	-0.997
Ackley	0.287	0.812	0.031	0.192	1.334	0.129	0.007	0.004	8.3e⁻⁴
Levy N. 13	0.053	0.959	0.021	0.050	2.197	0.028	7.9e ⁻⁴	5.5e⁻⁶	1.8e ⁻⁵
Styblinski-Tang	-75.127	-75.200	-68.614	5.878	4.123	5.981	-78.331	-78.332	-78.223

The surfaces of the benchmark functions used, are shown in the Fig. 11. Additionally, their mathematical representation for each one is given by Table 2, where it is also shown the global solutions and the search domains for each one. Those were the parameters use for the benchmark of the optimization methods.

To have the comparison of the optimization methods, every algorithm was implemented with a population of 40 particles for *PSO*, 40 bats for *BA* and 40 epicenters for the *EA*. Also, it is important to mention that the results of the three of them were evaluated after 100 iterations, and repeated 100 times.

To quantify the results of the algorithms, Table 3 shows the mean values obtained after taking the average of the 100 results of each algorithm after the 100 iterations.

Additional to that, the second group of results are for the calculated standard deviation, and finally the best solution found of each algorithm against every function is also shown, to validate their convergence property.

Also as it can be seen in Table 3, the standard deviation of the algorithms solutions are very constant, because except against the egg holder function, the *STD* of the algorithm emphasizes the constant performance of the algorithm.

The reason that increases the standard deviation of the algorithm against the egg holder function, is that searching area and the multiple local solutions, make the algorithm to sometimes look for a solution in an area that is out of the possible *Sr*, that is why above is recommended to search an adaptive way to improve the searching area to different problems.

On all cases presented in this work (benchmark functions, model optimization and controllers optimizations), the freedom grade given by the exponential distribution random generator, helped the algorithm to send a couple of epicenters to a fast search when it seemed that the group of epicenters started to look for a convergence in a local minimum.

Those heuristic “moves” are also fundamental for the performance of the algorithm, because the capabilities to “escape” from a local solution are improved. Therefore, the paper proves that the proposed algorithm works and that it can be used as an optimization method, knowing its sturdiness against different functions or applications.

5. Case study

As demonstrated in Section 4, the proposed Earthquake algorithm is able to reach solutions to different benchmark functions. However, in order to discuss the capability of the algorithm against a real application, a PID speed controller for a *DC* motor was implemented.

5.1. Implementation system

The experimental system designed for the tests consists of a DC motor with a 5V of nominal voltage, controlled by an H-Bridge [49] like the one shown in Fig. 12, which also shows the resultant circuit after activating the transistor Q_5 , where the red segmented line represents the activation current flow, and the green line the resultant energy leading to the motor movement.

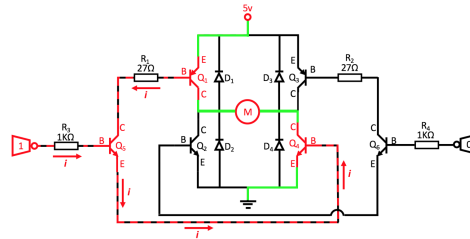


Figure 12. Schematic design of the implemented H-Bridge.

Additionally, in order to ensure the quality of the RPMs measurements of the DC motor a test environment was designed, where the motor could be fixed to a base, with a disc with a notch that together with an optical sensor of horseshoe type H21A1, allows to obtain a pulse counter per revolution. Fig. 13 shows the diagram of the testbed designed.

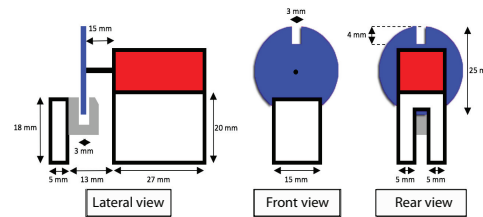


Figure 13. Testbed designed.

Thus, Fig. 13 shows the testbed designed to set it (motor in red color and components of the base in white), being that the shaft was secured to a disk of two millimeters thick (disk in blue), which passes through an optical horseshoe sensor (gray device) fix to the same base, for the measurement of its RPMs. The dimensions of the motor and optical sensor are found in [50] and [51] respectively.

As far as measurements are concerned, a low pass filter was implemented into the circuit according to input and output maximum values, 5 V and 5290 RPM, respectively:

$$f_{max} = \frac{5300 \text{ RPM} * 1 \text{ Hz}}{60 \text{ RPM}} = 83.33 \text{ Hz} \quad (8)$$

where f_{max} is the maximal frequency expected. Finally, the circuit schematic and board designs are given by Figs. 14 and 15 using the components in [50–55].

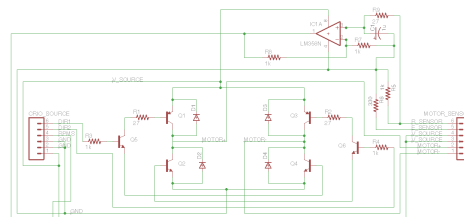


Figure 14. Schematic of the implemented test circuit.

In Fig. 14, the jacks connection at the left, are used for data acquisition and control signals. On the other hand, the pins at the right are used for the motor voltage and the optic sensor outputs.

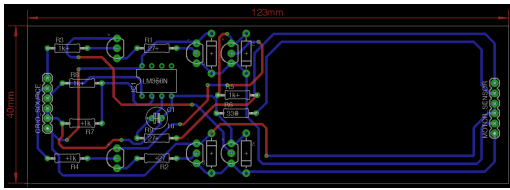


Figure 15. Board design of the implemented test circuit.

260 Fig. 15 shows the implemented circuit board design, taken from the schematic seen in figure
261 Fig. 14, which also clearly shows a space without components, same that is used to fix the motor base
262 of figure Fig. 13 to the circuit board.

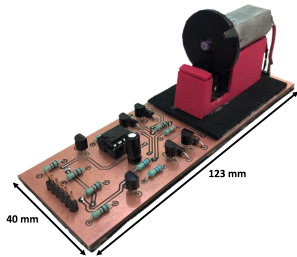


Figure 16. Testing environment.

263 The complete tests environment designed, is shown in Fig. 16, made for experimental speed
264 analysis and control, implemented with the PPN13KB DC motor [50].

265 5.2. Model optimization using EA, BA and PSO

266 The data acquisition system and the controllers implemented, where deployed in a FPGA NI
267 cRio-9068 [56] integrated with an I/O module NI 9381 [57], with a sample time of $6.85\mu\text{Sec}$. The
268 principle benefits of using an FPGA, relies on its performance because of the hardware parallelism
269 that it uses, moreover the reliability granted for avoiding the continually risk of a time-critical issue of
270 tasks preempting one another (which is a constant on software tools).

271 To obtain and later optimize the model of the DC motor, a step input from 0 to 5 volts was taken
272 to acquire the RPM response of the system, obtaining then the behavior shown Fig. 17, where can be
273 seen on the left axis the step input in percentage of duty cycle, and on the right the speed in RPMs,
274 both against time in seconds.

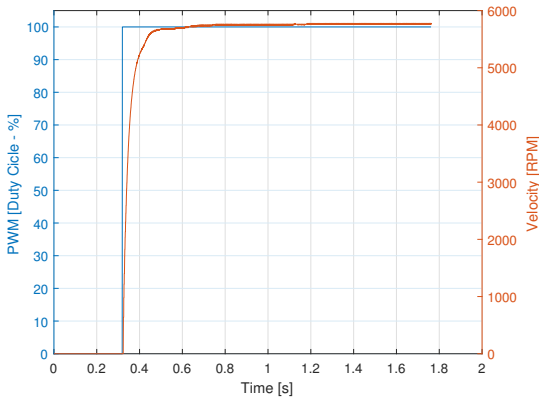


Figure 17. Experimental behavior of the DC motor.

Taking the data from Fig. 17, the a first experimental the continuous transfer function to be optimized (obtained by the Analytical Method (AM) [58]), is given by:

$$G(s) = \frac{2020}{s + 34.97} \quad (9)$$

And with a sample time $T = 6.85\mu S$, its discrete representation by:

$$G(z) = \frac{0.01384}{z - 0.9998} \quad (10)$$

To obtain a quantitative analysis of the control system performance (plant model and closed loop system) to evaluate and compared the improvement of the optimization algorithms implementation, performance indexes were selected to measure the dynamic and stationary error [59]. The improvement of the implementation were analyzed with four performance indexes:

1. Integral of the square of the error, *ISE*

$$ISE = \int_0^T e^2(t)dt \quad (11)$$

2. Integral of the absolute magnitude of the error, *IAE*

$$IAE = \int_0^T |e(t)|dt \quad (12)$$

3. Integral of time multiplied by absolute error, *ITAE*

$$ITAE = \int_0^T t \cdot |e(t)|dt \quad (13)$$

4. Integral of time multiplied by squared error, *ITSE*

$$ITSE = \int_0^T t \cdot e^2(t)dt \quad (14)$$

To improve the plant model obtained from analytic method, optimization algorithms are used and compared. Such algorithms are: (1) earthquake algorithm (EA), (2) bat algorithm (BA) and (3) particle swarm optimization (PSO).

The optimization process of the DC motor model is represented in Fig. 18, it shows how the EA is implemented to find the best parameters (gain and pole) with the objective to reduce the error from experimental data.

Also its important to mention that the cost function, that evaluates the performance index is based in the model of the *Integral of Time multiplied by the Absolute Error (ITAE)*, which properties lead the optimization to an absolute adjustment. The index is calculated with the discrete model of Eq. (10), which is calculated on every iteration with the *c2d* function of *MatlabTM*.

For the model optimization, each algorithm was implemented with a population of 40 particles for PSO, 40 bats for BA and 40 epicenters for EA, the three for 50 iterations, and repeated 10 times.

$$G(s) = \frac{Gain}{s + Pole} \quad (15)$$

The input model for every algorithm is given by the Eq. (15), and the results quantified in Table 4, where the best solution obtained from each algorithm is evaluated. Table 4 shows the values obtained for the tests.

Fig. 19 depicts the graphic behavior of the optimized transfer functions, where can be seen a comparison between the approximated models against the real measured data, all of them reacting to a step input.

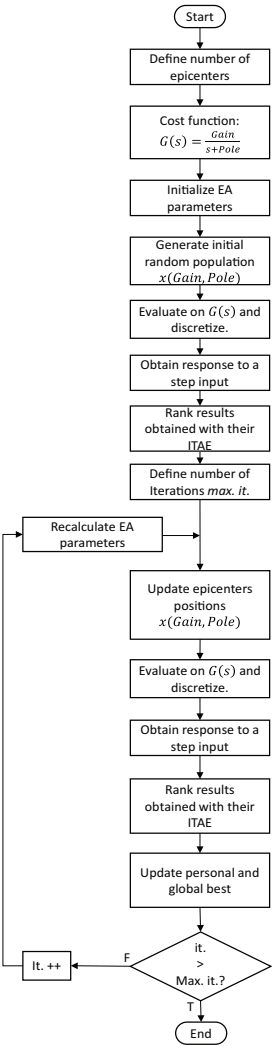


Figure 18. DC motor model optimization process.

Besides, Table 5 compares the performance indexes obtained for every algorithm, and also shows the indexes for the analytic method.

5.3. PID optimization using EA, BA and PSO

Knowing that the proposed study case, is the speed control for a DC motor, the controller implemented and optimized by EA, BA and PSO, is a parallel PID controller.

Then, from Fig. 20 the fitness function to optimize is given by its transfer function [58]:

$$U(t) = k_p e(t) + k_i \int e(t) dt + k_d \frac{de(t)}{dt} \tag{16}$$

where k_p , k_i and k_d are the proportional, integral and derivative gains, and $e(t)$ the error. Thus the non optimized values, obtained by the Ziegler-Nichols Method [58], are:

Table 4. Parameters obtained for plant models using each optimization algorithm.

Parameters	AM	EA	BA	PSO
Gain	2020	1961	2015	2019
Pole	34.97	34.12	35.05	35.14

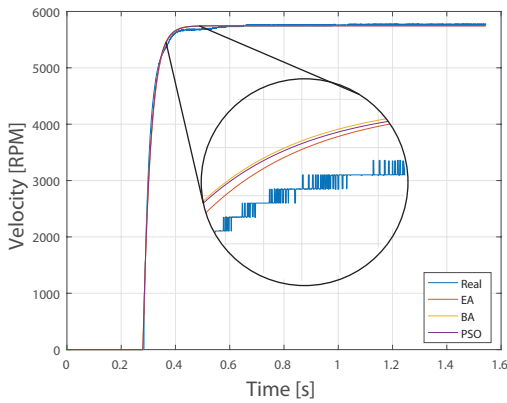


Figure 19. Transient response of DC motor models.

Table 5. Comparative table from indexes of the obtained plant models.

Indexes	AM	EA	BA	PSO
IAE	8.32e ⁶	6.22e ⁶	5.99e ⁶	6.39e ⁶
ISE	9.52e ⁸	5.90e ⁸	7.00e ⁸	7.00e ⁸
ITAE	56.96	41.04	42.60	43.81
ITSE	6.52e ³	4.04e ³	4.80e ³	4.80e ³

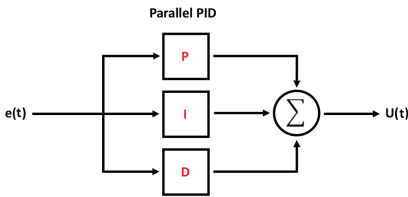


Figure 20. Blocks diagram of the classic PID parallel topology. [58]

- $k_p = 8e^{-6}$
- $k_i = 1e^{-6}$
- $k_d = 2e^{-8}$

Representation of the optimization process for tuning PID is visualized in Fig. 21 where the Eqs. (15) and (16) in closed loop are evaluated from each iterations obtaining the best parameters (k_p , k_i and k_d) reducing the dynamic error (ITAE) between the response and reference.

In order to strengthen the controller, and have another aspect to compare with the optimization algorithms, an adaptive PID controller using fuzzy logic was implemented. The topology implemented for the controller is given by Fig. 22.

As explained in [60], plotting the possible system inputs against the obtained outputs of the fuzzy sets, results in a control surface that can represent the entire set; reason why from the input and output sets (Fig. 23), evaluating with the proposed relation matrix (Table 6), the control surface of the system (Fig. 25) was obtained and implemented as show in Eq. (17).

The Fig. 24, shows how the mapping of the inputs against the outputs leads to the surface, which is finally obtained in Fig. 25. After obtaining the control surface, the curve is modeled to facilitate its implementation, adjusting it (as already mentioned) to the Eq. (17).

$$U(e) = \begin{cases} 0 \leq e < 0.25 & U = 1^{-8}e^3 - 5^{-9}e^2 + 1^{-9}e + 5^{-13} \\ 0.25 \leq e < 0.5 & U = 2^{-8}e^3 - 2^{-8}e^2 + 1^{-8}e - 2^{-9} \\ e > 0.5 & U = 6.86^{-10} \end{cases} \tag{17}$$

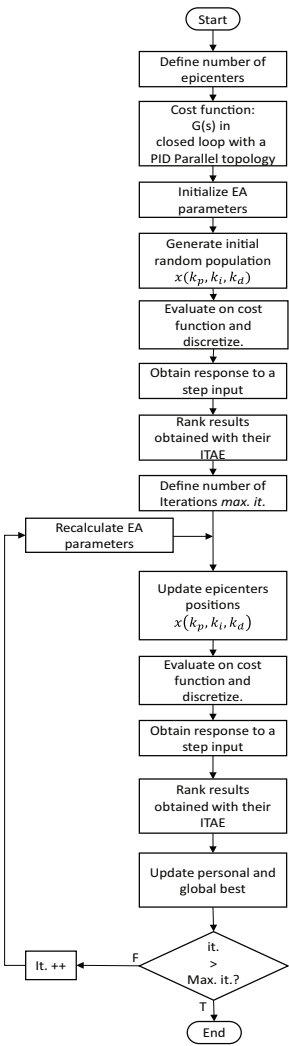


Figure 21. PID tuning optimization process.

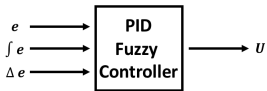


Figure 22. Blocks diagram implemented for the fuzzy PID Mamdani type [60].

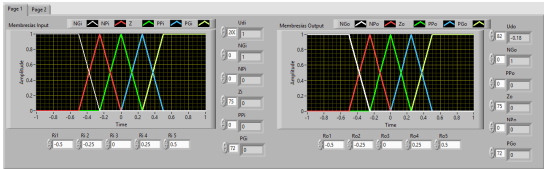


Figure 23. The fuzzy sets implemented for the Mamdani fuzzy PID controller.

325 where $U(e)$ is the control surface, evaluated on the error e .
326 The general structure implemented for the PID and the Fuzzy- PID control system, can be resumed
327 as shown in Fig. 26, where in the HMI Host was implemented the Fuzzy configuration to estimate the
328 surface between inputs/outputs, to optimize the PID controller embedded in the $cRio$ $FPGA$.

Table 6. Relation matrix for the input and output fuzzy sets.

	e				
	NG_{out}	NP_{out}	Z_{out}	PP_{out}	PG_{out}
NG_{in}	1	0	0	0	0
NP_{in}	0	1	0	0	0
Z_{in}	0	0	1	0	0
PP_{in}	0	0	0	1	0
PG_{in}	0	0	0	0	1

	$\int e$				
NG_{in}	0	0	0	0	0
NP_{in}	0	1	0	0	0
Z_{in}	0	0	1	0	0
PP_{in}	0	0	0	1	0
PG_{in}	0	0	0	0	0

	Δe				
NG_{in}	1	0	0	0	0
NP_{in}	0	1	0	0	0
Z_{in}	0	0	1	0	0
PP_{in}	0	0	0	1	0
PG_{in}	0	0	0	0	1

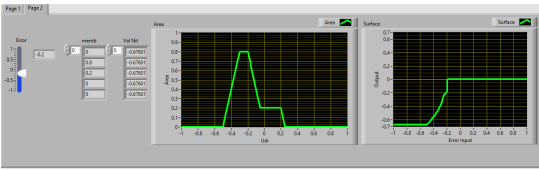


Figure 24. Mapping process.

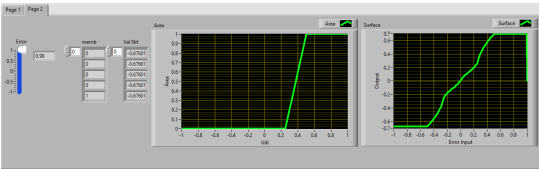


Figure 25. Control surface.

329 Finally, the optimization algorithms already mentioned where implemented using the same
330 population, iterations and repetitions as in Section 5.2. Nevertheless, in this section the algorithms
331 were used first to obtain randomly a transfer function, and then its corresponding *PID* constants.

332 Taking the same criteria as in the model optimization, the selected constants were taken from the
333 best set of solutions. Table 7, shows the values obtained for the tests.

334 The Fig. 27 shows a comparative between the behaviors of the implemented methods, where the
335 *PID* graph represents the non-optimized *PID*, and the reference plot shows the input step, where the
336 reference is placed on 4000 RPM. On the other hand, the quantification is given by Table 8, that shows
337 the performance indexes of every method.

338 Evaluating the obtained parameters on a adaptive controller, Fig. 28 shows a comparative between
339 the behaviors of the implemented methods in a fuzzy *PID Mamdani* type controller, where the *PID*
340 graph also represents the non-optimized fuzzy *PID*, and the reference plot shows the input step, where
341 the reference also for this case is placed on 4000 RPM. On the other hand, the quantification is given by
342 Table 9, that shows the performance indexes of every method.

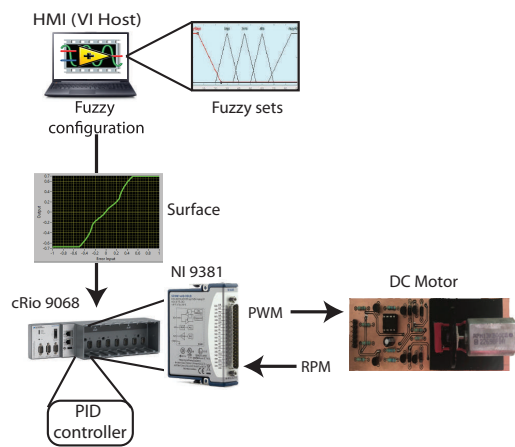


Figure 26. General structure of the control System implemented

Table 7. PID constants obtained for plant models using each optimization algorithm.

Gains	EA	BA	PSO
k_p	$0.1162e^{-4}$	$0.3536e^{-4}$	$0.1711e^{-4}$
k_i	$0.0907e^{-4}$	$0.0476e^{-4}$	$0.0235e^{-4}$
k_d	$0.0040e^{-4}$	$0.0059e^{-4}$	$0.0008e^{-4}$

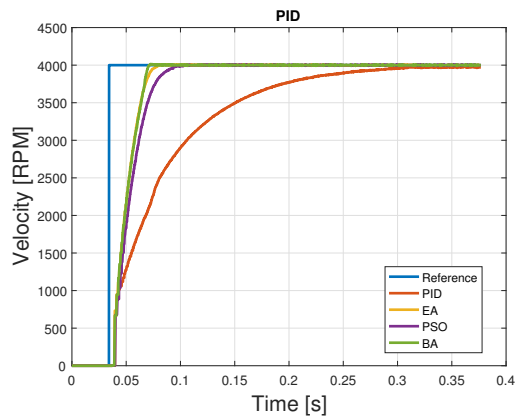


Figure 27. Comparisons of transient responses to step change in the reference with conventional PID structure.

Nevertheless, the cost function for this section is also evaluated with the performance index based in the model of the *Integral of Time multiplied by the Absolute Error (ITAE)*, which properties lead the optimization to an absolute adjustment. Resuming in every iteration the global best as the epicenter with the minimal *ITAE*.

6. Discussion

The Earthquake Algorithm presented in this paper, was tested against two of the most implemented optimization algorithms (*PSO* and *BA*), in order to prove and compare how the *EA* solves optimization problems.

As most of the modern metaheuristic optimization methods, the *EA* is inspired on a behavior that exists in the nature, its strengths are found in the capability of using two kinds of velocity to find a solution. In section [Section 4](#), the method was tested with different benchmark functions,

Table 8. Comparative table from indexes on conventional *PID* structure.

Indexes	PID	EA	BA	PSO
IAE	3.28e ⁷	9.87e ⁶	9.97e ⁶	1.19e ⁷
ISE	5.68e ¹⁰	2.59e ¹⁰	2.61e ¹⁰	3.03e ¹⁰
ITAE	224.79	67.62	68.29	81.73
ITSE	3.89e ⁵	1.78e ⁵	1.79e ⁵	2.07e ⁵

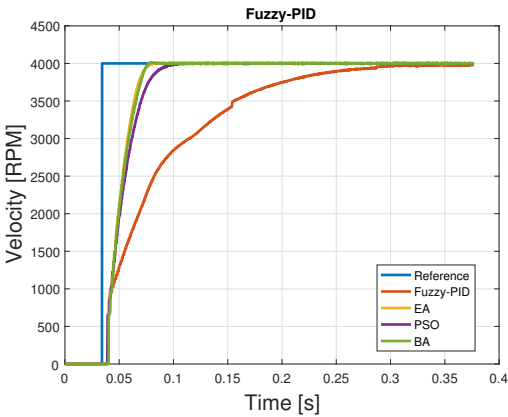


Figure 28. Comparisons of transient responses to step change with Fuzzy *PID* structure.

demonstrating to have the best performance against the Keane function, Rastrigin function and the Holder table, compared to the *BA* and *PSO*.

Those functions share that they are continuous, non-separable, multimodal, and somehow that their global minimums are surrounded by local solutions. The difference between them and functions like Schaffer n.2 or Levy, is in the separation of the possible solutions, because the Earthquake Algorithm proved to have a better performance that the other two, in cases where it can exploit more its versatility to adapt speed for distant searches and subsequently nearby.

Talking about convergence, the *PSO* showed that for functions like the Sphere function it has the best performance, but after adding relief (like in the Keane function) the *PSO* reduces its performance and the *EA* shows a better or equal performance. That property, is improved in the *EA* by the v_s parameter, that allows the algorithm to do a finer search when needed.

However, against the other benchmark functions, the algorithm also had acceptable performances, seeing that the *EA* never had the worse mean result against any of the functions. Actually, the egg holder function was the most difficult test for the algorithm, though even that function was perfectly solved by the algorithm a couple of times.

Analyzing the Egg holder function, the *BA* clearly has the beast mean solution, but the *EA* also found great solutions but with a bigger standard deviation. That could be improved, with an adaptive range for the random generation with the exponential distribution, to help the algorithm prevent falling into a local minimum on a bigger range. Here (in the egg holder function) or against the table holder, the Keane or the Rastrigin functions, the v_p parameter shows its strength, because that is the

Table 9. Comparative table from indexes on Fuzzy *PID* structure.

Indexes	PID	EA	BA	PSO
IAE	3.44e ⁷	1.02e ⁷	1.05e ⁷	1.18e ⁷
ISE	5.91e ¹⁰	2.71e ¹⁰	2.80e ¹⁰	2.89e ¹⁰
ITAE	235.40	70.11	71.94	80.85
ITSE	4.05e ⁵	1.86e ⁵	1.92e ⁵	1.98e ⁵

parameter that allow the algorithm to make quick searches at greater distances, to finally switch back to fine searches.

Besides, in the study case where a speed control was implemented and improved for a *DC* motor, it has to be considered that the "calibration" of the optimization was made based on the *ITAE*, reason why in the tables of [Section 5](#) it is shown that the responses fits better to that index.

On the other hand, the obtained results, show that the algorithm can also be implemented for an unconstrained search of solutions. Referring then to the *ITAE*, the founded solutions of the optimization algorithms did not show big differences, but in the three cases (model optimization, *PID* implementation and the adaptive controller) the *EA* showed a lightly better performance, that the other two algorithms.

For the controllers designs, the *BA* and the *EA* had similar behaviors, but both had better solutions against the *PSO* optimization. In the model optimization, it can be seen that the indexes are more even than the gains indexes.

Therefore, looking at that index in [Tables 5, 8 and 9](#), is clear that (maybe with not a great difference) the best solutions quantified are given by the Earthquake Algorithm.

By the way, the *EA* still has improvement opportunities, which mainly consist on the expression that corresponds to the *Sr*, because making the *S range* adaptable to every problem or situation could improve the searching performance.

7. Conclusions

The proposed algorithm in this paper, opens a new investigation line with a Geo-inspired metaheuristic algorithm.

Nevertheless, it is important to highlight the results obtained against the benchmark functions, because it clearly highlights the constancy and stability of the method, by showing that it is often close to the best solutions, knowing that the *EA* obtained the best mean solution against the table holder function, the Keane function and the Rastrigin function. Perhaps it did not have the best solution on every benchmark function (comparing it to *PSO* and *BA*), it always had a good performance leaving aside that the algorithm showed that it never went far from the global minimums. In other words, results proved that between the three algorithms, *EA* had meanly the best solutions according to [Table 3](#).

The implementation of two speed possibilities for the search, and also the exponential distribution implemented for random epicenters, make the *EA* more robust against the possibility of falling into a local solution near to the global best. That property is mainly shown in the table holder function, in the Keane function and in the Rastrigin function.

From engineering application results, the optimization using *EA* for both modeling and *PID*-tuning proved that the speed control system improved the dynamic response when tracking changes are presented because the optimization was based on *ITAE* which objective is reduce the dynamic error between speed of reference and response. Optimization made by *EA* presented a better speed control than *BA* and *PSO* algorithms in reference tracking behavior.

Author Contributions: The conceptualization, experimentation and software are completed by E.M. and A.O. The visualization, formal analysis and supervision are directed by P.P. and A.M. instructs the project administration.

Conflicts of Interest: The authors declare no conflict of interest.

1. Yang, X. *Nature-inspired metaheuristic algorithms*, 2. ed ed.; Luniver Press: Frome, 2010. OCLC: 1007278873.
2. Ebrahimi, A.; Khamsehchi, E. Sperm whale algorithm: An effective metaheuristic algorithm for production optimization problems. *Journal of Natural Gas Science and Engineering* **2016**, *29*, 211–222.
3. Rangaiah, G.P. *Stochastic Global Optimization: Techniques and Applications in Chemical Engineering*; World Scientific Publishing Co., Inc.: River Edge, NJ, USA, 2010.

4. Askarzadeh, A. A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm. *Computers & Structures* **2016**, *169*, 1–12.
5. Ribeiro, C.C.; Martins, S.L.; Rosseti, I. Metaheuristics for optimization problems in computer communications. *Computer Communications* **2007**, *30*, 656–669.
6. Nanda, S.J.; Panda, G. A survey on nature inspired metaheuristic algorithms for partitional clustering. *Swarm and Evolutionary Computation* **2014**, *16*, 1–18.
7. Boussaïd, I.; Lepagnot, J.; Siarry, P. A survey on optimization metaheuristics. *Information Sciences* **2013**, *237*, 82–117.
8. Blum, C.; Roli, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM computing surveys (CSUR)* **2003**, *35*, 268–308.
9. Nematollahi, A.F.; Rahiminejad, A.; Vahidi, B. A novel physical based meta-heuristic optimization method known as Lightning Attachment Procedure Optimization. *Applied Soft Computing* **2017**, *59*, 596 – 621.
10. Elbeltagi, E.; Hegazy, T.; Grierson, D. Comparison among five evolutionary-based optimization algorithms. *Advanced engineering informatics* **2005**, *19*, 43–53.
11. Holland, J.H. Adaptation in natural and artificial systems. An introductory analysis with application to biology, control, and artificial intelligence. *Ann Arbor, MI: University of Michigan Press* **1975**, pp. 439–444.
12. Fogel, L.J.; Owens, A.J.; Walsh, M.J. Artificial intelligence through simulated evolution **1966**.
13. Rechenberg, I. Cybernetic solution path of an experimental problem **1965**.
14. Storn, R.; Price, K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization* **1997**, *11*, 341–359.
15. Dhiman, G.; Kumar, V. Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications. *Advances in Engineering Software* **2017**, *114*, 48–70.
16. Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P. Optimization by simulated annealing. *science* **1983**, *220*, 671–680.
17. Rashedi, E.; Nezamabadi-pour, H.; Saryazdi, S. GSA: A Gravitational Search Algorithm. *Information Sciences* **2009**, *179*, 2232 – 2248. Special Section on High Order Fuzzy Sets.
18. Tayarani-N, M.H.; Akbarzadeh-T, M. Magnetic optimization algorithms a new synthesis. *Evolutionary Computation*, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on. IEEE, 2008, pp. 2659–2664.
19. Kaveh, A.; Talatahari, S. A novel heuristic optimization method: charged system search. *Acta Mechanica* **2010**, *213*, 267–289.
20. Hatamlou, A. Black hole: A new heuristic optimization approach for data clustering. *Information sciences* **2013**, *222*, 175–184.
21. Shah-Hosseini, H. Principal components analysis by the galaxy-based search algorithm: a novel metaheuristic for continuous optimisation. *International Journal of Computational Science and Engineering* **2011**, *6*, 132–140.
22. Alatas, B. ACROA: Artificial Chemical Reaction Optimization Algorithm for global optimization. *Expert Systems with Applications* **2011**, *38*, 13170 – 13180.
23. Du, H.; Wu, X.; Zhuang, J. Small-world optimization algorithm for function optimization. *International Conference on Natural Computation*. Springer, 2006, pp. 264–273.
24. Yang, X.S.; Deb, S.; Fong, S.; He, X.; Zhao, Y.X. From Swarm Intelligence to Metaheuristics: Nature-Inspired Optimization Algorithms. *Computer* **2016**, *49*, 52–59.
25. Eberhart, R.; Kennedy, J. A new optimizer using particle swarm theory. *Micro Machine and Human Science*, 1995. MHS'95., Proceedings of the Sixth International Symposium on. IEEE, 1995, pp. 39–43.
26. Mucherino, A.; Seref, O. Monkey search: a novel metaheuristic search for global optimization. *AIP conference proceedings*. AIP, 2007, Vol. 953, pp. 162–173.
27. Yang, X.S. A new metaheuristic bat-inspired algorithm. In *Nature inspired cooperative strategies for optimization (NICSO 2010)*; Springer, 2010; pp. 65–74.
28. Gandomi, A.H.; Yang, X.S.; Alavi, A.H. Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Engineering with Computers* **2013**, *29*, 17–35.
29. Yang, C.; Tu, X.; Chen, J. Algorithm of marriage in honey bees optimization based on the wolf pack search. *Intelligent Pervasive Computing*, 2007. IPC. The 2007 International Conference on. IEEE, 2007, pp. 462–467.

- 473 30. Yang, X.S. Firefly algorithms for multimodal optimization. International symposium on stochastic
474 algorithms. Springer, 2009, pp. 169–178.
- 475 31. Karaboga, D.; Basturk, B. A powerful and efficient algorithm for numerical function optimization: artificial
476 bee colony (ABC) algorithm. *Journal of global optimization* **2007**, *39*, 459–471.
- 477 32. Dorigo, M.; Birattari, M. Ant colony optimization. In *Encyclopedia of machine learning*; Springer, 2011; pp.
478 36–39.
- 479 33. Rao, R.; Savsani, V.; Vakharia, D. Teaching–learning-based optimization: A novel method for constrained
480 mechanical design optimization problems. *Computer-Aided Design* **2011**, *43*, 303 – 315.
- 481 34. Glover, F. Tabu Search, Part I. *ORSA Journal on Computing* **1989**, *1*, 190–206.
- 482 35. Talatahari, S.; Azar, B.F.; Sheikholeslami, R.; Gandomi, A. Imperialist competitive algorithm combined
483 with chaos for global optimization. *Communications in Nonlinear Science and Numerical Simulation* **2012**,
484 *17*, 1312 – 1319.
- 485 36. Geem, Z.W.; Kim, J.H.; Loganathan, G.V. A new heuristic optimization algorithm: harmony search.
486 *Simulation* **2001**, *76*, 60–68.
- 487 37. Kashan, A.H. League Championship Algorithm (LCA): An algorithm for global optimization inspired by
488 sport championships. *Applied Soft Computing* **2014**, *16*, 171 – 200.
- 489 38. Karaboga, T.; Canyilmaz, M.; Ozcan, O. Investigation of the relationship between ionospheric foF2 and
490 earthquakes. *Advances in Space Research* **2018**, *61*, 2022–2030.
- 491 39. Lay, T. A review of the rupture characteristics of the 2011 Tohoku-oki Mw 9.1 earthquake. *Tectonophysics*
492 **2017**.
- 493 40. Olson, E.L.; Allen, R.M. The deterministic nature of earthquake rupture. *Nature* **2005**, *438*, 212.
- 494 41. Bormann, P. *New Manual of Seismological Observatory Practice (NMSOP-2)*; Vol. 2 Volumes, 2002.
- 495 42. Bormann, P.; Engdahl, B.; Rainer, K. Seismic Wave Propagation and Earth models **2009**.
- 496 43. Allaby, M. *Dictionary of Geology and Earth Sciences*; Oxford University Press, 2013.
- 497 44. UNAM. Seismic data provided by the strong ground motion database system.
498 url(<http://aplicaciones.iingen.unam.mx/AcelerogramasRSM/>), 2014. Accessed in feb-2018.
- 499 45. Ponce, P.; Molina, A. Internal research report Binational-laboratory. Technical report, Tecnologico de
500 Monterrey, Mexico, 2018.
- 501 46. Cooper, J.C. The poisson and exponential distributions. *Mathematical Spectrum* **2005**, *37*, 123–125.
- 502 47. Jamil, M.; Yang, X.S. A literature survey of benchmark functions for global optimisation problems.
503 *International Journal of Mathematical Modelling and Numerical Optimisation* **2013**, *4*, 150–194.
- 504 48. Molga, M.; Smutnicki, C. Test functions for optimization needs. *Test functions for optimization needs* **2005**,
505 *101*.
- 506 49. Khoucha, F.; Lagoun, S.M.; Marouani, K.; Kheloui, A.; Benbouzid, M.E.H. Hybrid cascaded H-bridge
507 multilevel-inverter induction-motor-drive direct torque control for automotive applications. *IEEE*
508 *Transactions on Industrial Electronics* **2010**, *57*, 892–899.
- 509 50. Ltd., M.C. DC Brush Motor PPN13, 2010.
- 510 51. Semiconductors, F. H22A1 Slotted Optical Switch, 2010.
- 511 52. Instruments, T. LMx58-N Low-Power, Dual-Operational Amplifiers, 2014.
- 512 53. Semiconductor, F. 2n3904/mmbt3904/pzt3904 npn general purpose amplifier, 2014.
- 513 54. Semiconductor, F. 2N3906 Data Sheet. *South Portland, ME* **2000**.
- 514 55. Semiconductor, F. 1N4001-4007 General Purpose Rectifiers. *Online Document* **2009**.
- 515 56. NI. NI cRIO-9068 Reconfigurable Embedded Chassis with Integrated Intelligent RealTime Controller for
516 CompactRIO, USER MANUAL, 2016.
- 517 57. NI. NI 9381, 8 AI/8 AO/4 DIO, 0 V to 5 V, 12 Bit, 20 kS/s Aggregate, DATASHEET, 2016.
- 518 58. Ogata, K.; Yang, Y. *Modern control engineering*; Vol. 4, Prentice hall India, 2002.
- 519 59. Gozde, H.; Taplamacioglu, M.C.; İlhan Kocaarslan. Comparative performance analysis of Artificial
520 Bee Colony algorithm in automatic generation control for interconnected reheat thermal power system.
521 *International Journal of Electrical Power & Energy Systems* **2012**, *42*, 167 – 178.
- 522 60. Cruz, P.P.; Herrera, A. *Inteligencia artificial con aplicaciones a la ingeniería*; Vol. 1, Marcombo, 2011.