Application of Cellular Automatons and Genetic Algorithms in the Creation of a Steganographic Technique

Aplicación de Autómatas Celulares y Algoritmos Genéticos en la Creación de una Técnica Esteganográfica

Lilia Edith Aparicio Pico[1] , Árnol Guerrero Calderón[2]

Abstract

This article describes the creation of a Steganographic technique focused on image analysis, supported by Artificial Intelligence (AI) methods and mathematical optimization models [18]. Initially, a two-dimensional Cellular Automaton (AC) (d=2) with Moore's vicinity (N(1)=8) is used, evolution is semitotalistic in order to select the most suitable pixels (cells), from those defined in Eden, by applying a set of transition rules, defined in terms of the Game of Life, over several generations. Then through Genetic Algorithms the CA rules will be improved, to finally obtain a position vector, which will be given by the implementation of a one-dimensional AC, whose operation is executed using the rules of Wolfram 30 or 150 as the case may be [1]. This method has been called Modular Steganography.

Keywords: Cellular automaton, Evolution, Genetic algorithm, Security, Steganography.

Resumen

El presente artículo describe la creación de una técnica Esteganográfica enfocada en el análisis de imágenes, soportada en métodos de Inteligencia Artificial (IA) y modelos

[1] Bachelor of Science in Education, Universidad Distrital Francisco José de Caldas, Colombia. Doctorate in Technical Sciences, Universidad Central Marta Abreu De Las Villas, Cuba. Staff Professor: Universidad Distrital Francisco José de Caldas, Colombia. E-mail: eaparicio@udistrital.edu.co ORCID: https://orcid.org/0000-0003-1841-4423

[2] Systems Engineering, Universidad Autónoma de Colombia, Colombia. Master in Information and Communications Sciences, Universidad Distrital Francisco José de Caldas, Colombia. CEO: Cuántica SAS, Colombia. E-mail: arguerreroc@correo.udistrital.edu.co ORCID: https://orcid.org/0000-0002-8310-810X

matemáticos de optimización [18]. Inicialmente, se utiliza un Autómata Celular (AC) bidimensional (d=2) con vecindad de Moore (N(1)=8), la evolución es semitotalista con el fin de seleccionar los píxeles (células) más aptos, de aquellos definidos en el Edén, mediante la aplicación de un conjunto de reglas de transición, definidas en términos del Juego de la Vida, a través de varias generaciones. Luego mediante Algoritmos Genéticos se mejorarán las reglas del AC, para finalmente obtener un vector de posición, que estará dado por la implementación de un AC unidimensional, cuya operación se ejecuta mediante las reglas de Wólfram 30 o 150 según el caso [1]. Este método ha sido denominado Esteganografía Modular.

Palabras clave: Autómata celular, Evolución, Algoritmo genético, Seguridad, Esteganografía.

1. Introduction

One of the most striking techniques in the context of information security is steganography. The control of bits, within images, sound and video, emerges as an alternative of this computational science when hiding information within digital files [16]. For the case study, the treatment of digital images will be detailed through a holistic approach analyzing the step by step of the whole research.

The application of Cellular Automata (CA's) that evolve with Genetic Algorithms (GA's) [14] is suggested, giving rise to Evolutionary Genetic Automata. The model arises from merging GA's and CA's dazzling the possibility of using these tools in Information Security implementations. With the use of a two-dimensional AC [2], whose development is detailed in sections 2 and 3, it will be determined which pixels of an image are suitable or not (alive or dead) to be included in the list of pixels subject to be modified with the abstraction of their values and the subsequent inclusion of data, given the evaluation of neighbor states; after several generations and after verifying, by emulating the LSB technique, that the number of selected pixels is not sufficient, a GA will be implemented to improve the rules predisposed for the Automaton.

This procedure is described in section 4. Modular Steganography, proposes through ordered coordinate referencing an abstraction of the values of each pixel by implementing a one-dimensional CA. This technique has been named SteganoS-AA since it allows the interaction of Cellular Automata and Genetic Algorithms in the complex process of data hiding, it is framed to discover other potential applications of Genetic Algorithms and in turn, to implement a mathematical model of optimization, the Cellular Automata, in the creation of a method of data hiding, under the theories proposed from Von Neumann to Stephen Wolfram.

2. Cellular Automaton Procedure

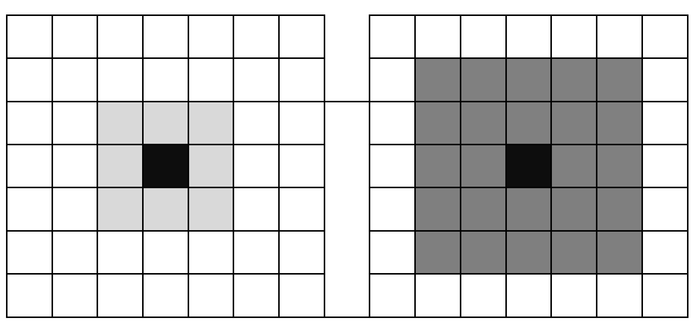
The starting point through time from which the automaton begins to evolve is defined by the following initial conditions, also called "Eden":

$$A\ two - dimensional\ Cellular\ Automaton\ (CA)\ is\ used\ (d = 2)$$
$$with\ Moore\ neighborhood\ (N(1) = 8)$$

$$AC(L, S, N, f) =$$
$$AC(Image\ Dimension, \{0,1\}, N(1) = 8, GOL)$$
$$random\ eden, where:$$

$$L\ is\ the\ lattice\ region\ of\ dimension\ d, where\ the\ elements$$
$$of\ L\ are\ called\ cells\ (pixels),$$
$$S\ is\ the\ finite\ set\ of\ states,$$
$$N\ is\ the\ finite\ set\ of\ neighborhoods\ and$$
$$f\ is\ the\ transition\ function\ f: S^N \rightarrow S$$

Figure 1. Moore's neighborhood r. a) Traditional, r = 1, b) Extended r = 2. [10].



a) b)

The lattice region of dimension $d = 2$ represents a planar surface formed by a finite number of cells based on the following configurations, where $i = 0\ j = 0$:

$$\begin{bmatrix} \chi_{i-2,j-2} & \chi_{i-2,j-1} & \chi_{i-2,j} & \chi_{i-2,j+1} & \chi_{i-2,j+2} \\ \chi_{i-1,j-2} & \chi_{i-1,j-1} & \chi_{i-1,j} & \chi_{i-1,j+1} & \chi_{i-1,j+2} \\ \chi_{i,j-2} & \chi_{i,j-1} & \chi_{ij} & \chi_{i,j+1} & \chi_{i,j+2} \\ \chi_{i+1,j-2} & \chi_{i+1,j-1} & \chi_{i+1,j} & \chi_{i+1,j+1} & \chi_{i+1,j+2} \\ \chi_{i+2,j-2} & \chi_{i+2,j-1} & \chi_{i+2,j} & \chi_{i+2,j+1} & \chi_{i+2,j+2} \end{bmatrix}$$

The transition function f uses Moore neighborhood (Figure 1), the Moore neighborhood consists of a central cell and neighbors around it, such that $|V| = (2r + 1)^2 - 1$where r represents the neighborhood radius. For the case of study, we would have $((2 * (1)) + 1)^2 - 1 = 8\, neighborhoods$ [3]. The initial rule of evolution is semitotalistic and has been represented as R(2,3,3,3,3), the first pair of numbers 2,3 are a minimum and a maximum respectively of living cells necessary to define the survival of a cell that is alive, the second pair of numbers 3,3 are also a minimum and a maximum respectively of living cells necessary to define the birth of a new cell. Then, the variables take the corresponding values, $S_{min} = 2, S_{max} = 3, B_{min} = 3,\ B_{max} = 3,$ such that S represents the survival over time $t + 1$ of a cell that is alive at time t y B represents the rebirth of a cell in time of a cell that was dead in time. $t + 1$ of a cell that had been dead in time. t. The initial rule, Conway's rule of life [4], can be expressed as follows:

THE PIXELS will remain alive in time $t + 1$if in $r = 1$if, if and only if, two or three neighboring pixels are alive in time $t$otherwise they will die.

THE PIXELS will be resurrected (they are dead) for $time\, t + 1$if in $r = 1$if, if and only if, three neighboring pixels are alive in $time\, t$

$$\boldsymbol{Transition\ function} = f(S23/B3)$$

The equation for the R (2,3,3,3,3) rule is as follows:

$$(x_0, x_1, \dots., x_7) = \begin{cases} 1, \quad si \quad \begin{cases} x_0 = 0 \quad y \quad 3 \leq \sum_{i=1}^{8} x_i \leq 3 \\ x_0 = 1 \quad y \quad 2 \leq \sum_{i=1}^{8} x_i \leq 3 \end{cases} \\ 0 \ \ in\ another\ case \end{cases}$$

It follows that $3 \leq \sum_{i=1}^{8} x_i \leq 3$ is the interval of live cells between $B_{min}\ y\ B_{max}$ for a birth and $2 \leq \sum_{i=1}^{8} x_i \leq 3$ is the interval of live cells between $S_{min}\ y\ S_{max}$ for survival.

After 500 generations, and according to the analysis of surviving pixels, it is decided whether to start the information embedding process, otherwise, the Genetic Algorithm procedure will evolve the rules for a new evolutionary process.

3. Implementation of the Cellular Automaton in a case of application.

The exercise starts from the selection of "*Eden*". The initial cells (pixels) are selected from the image to be treated, by calculating the local gradient corresponding to each pixel of the image, to determine the edges (sharp areas) that will focus the primary selection of the most suitable pixels.

The *Sobel* edge operator [5] is used to calculate the gradient. This operator has a filter defined by the following coefficient matrix:

$$H_x^S = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \; y \; H_y^S = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

The result of the *Sobel* filter produces estimates of the local gradient for all pixels of the image in its two different directions, maintaining the following relationship:

$$\nabla I(x,y) \approx \frac{1}{8} \begin{bmatrix} H_y^S & \cdot I \\ H_y^S & \cdot I \end{bmatrix}$$

The filter result is now characterized as,

$$D_x(x,y) = H_x * I \; y \; D_y(x,y) = H_y * I$$

The magnitude of the edge $E(u,v)$ is defined as the magnitude of the gradient:

$$E(x,y) = \sqrt{(D_x(x,y))^2 + (D_y(x,y))^2}$$

and the direction of the gradient in each pixel (the angle) is calculated from:

$$\phi(x,y) = tan^{-1} \left[\frac{D_y(x,y)}{D_x(x,y)} \right]$$

The whole process for edge detection is summarized by filtering the original image through the two coefficient matrices H_x y H_y and consequently their results are gathered on the magnitude of the gradient $E(x,y)$ and the direction of the gradient $\phi(x,y)$ [5].

The edge function of the image processing toolbox in MATLAB will be used, this function allows to implement the *Sobel* edge operator. First the image named *'RapaNui.bmp'* is read and placed in the variable 'I', then in the variable 'A' the same image is loaded, but converted to gray using the *rgb2gray* function. Finally, the edge operator *'Sobel'* is implemented, obtaining the following result:

Figure 1. a) Original RapaNui image (1287 x 2048). b) Image with defined edges.

a) b)

Source: Own

The details of Image 1 b) can be reviewed in Table 1. It provides information on the initial population for time $t = 0$: BINARIZED.

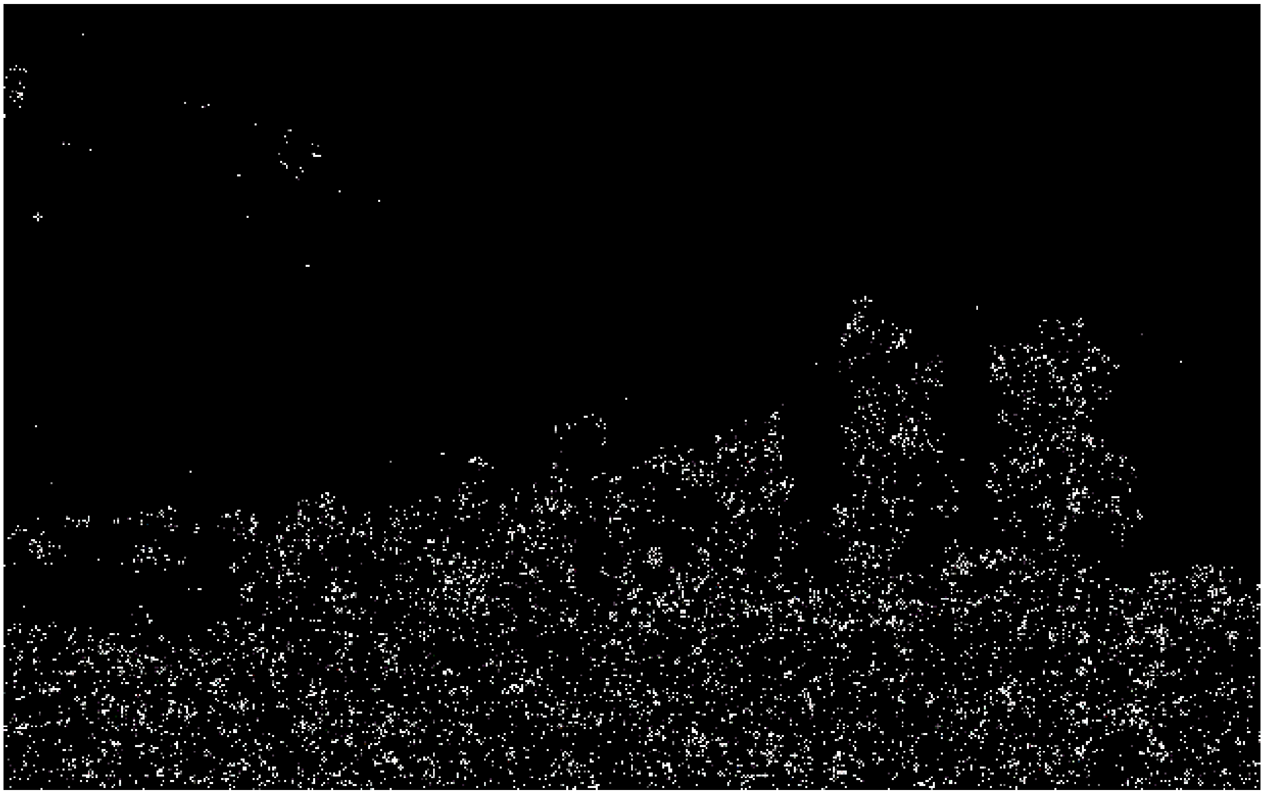
Table 1. Initial pixel count before applying GOL (Game Of Life).

Initial Pixel Status	Number of Pixels	Percentage of pixels in relation to the total image
Alive (1)	109.048	4,13%
Dead (0)	2.526.728	95,87%

Then the algorithm elaborated to implement the Conway rule of life R(2,3,3,3,3) [4] is executed, it has been established that for time $t = 500$ generations[3] the efficiency is evaluated with the transition function, as illustrated in Figure 2:

[3]. We determined 500 generations, since during the tests performed on more than 80 images, the transformation rate, generation after generation, did not undergo profound changes after reaching 500 generations.

Figure 2. Pixel population after 500 generations**.**



Source: Own

Table 2. Detail of pixel population for $t = 500$ generations.

Initial Pixel Status	Number of Pixels	Percentage of pixels in relation to the total image
Alive (1)	47.194	2%
Dead (0)	2.588.582	98%

At this point of the CA procedure, Table 3 details the analysis, prior execution of the GA procedure:

Table 3. Detail of available bits if the LSB replacement technique is applied.

Total pixels of treated image	Pixels for t=500	Available LSB-replacement bits (1)	Available LSB-replacement bits (2)	Available LSB-replacement bits (3)	
2.635.776	47.194 (2%)	141.582	283.164	424.746	bits
		17.697,75	35.395,50	53.093,25	bytes
		17,28	34,57	51,85	Kb
		0,02	0,03	0,05	Mb

Once known, at this initial stage, the available pixels (survivors), the LSB (Least Significant Bit)-replacement technique [13] can be applied in its variant established for this proposal, based on the selection function, which consists of extracting the least significant bit or bits (the rightmost of each byte of each pixel at 24 bits) of each pixel and comparing it, bit by bit, with the message to be hidden, previously transformed to binary. If a difference is detected the least significant bit or bits of that pixel are modified. If it is a bit '0' it is converted to a bit '1' and if it is a bit '1' to a bit '0', as the case may be [6]. The insertion technique will be detailed later in this section.

Table 3, in its last three columns, details the number of bits available, for each case of the least significant bit or bits of the pixels available for the insertion of the hidden message. If the size in bits of the message to be hidden exceeds the capacity (selection threshold) set for time $t = 500$The rule of the GOL life of the AC must undergo an evolution through the application of the GA, which is detailed in the following item.

4. **Procedure for the Genetic Algorithm**

The optimization process starts with the "initialization", which consists in the generation of a population of individuals [7][15]. The initial population will be obtained, according to the detail of the initial rules set for the original CA. Therefore, the population will be represented by the set of rules of the CA. Each subset of rules defines a specific task to be performed by the automaton. The starting point will be the population generated, after a number of generations by the CA and that have not exceeded the selection threshold.

Then an "evaluation" exercise is performed in which each subset of rules (population of individuals) is subjected to an efficiency or fitness function. Of course, the objective of the fitness function will be to enhance the effect of the selection rules on the CA's behavior.

1. For the specific case of the proposal, the initial population for the GA is established as the individuals derived from the initial rule, R(2,3,3,3,3), of the CA evolution process, as shown in Table 4.

Table 4. Binary detail of the R(2,3,3,3,3) rule, initial population.

	Initial Population			
A	**0 0 0 0 0 0 1 0**	**0 0 0 0 0 0 1 1**	**0 0 0 0 0 0 1 1**	**0 0 0 0 0 0 1 1**
	2	**3**	3	3
	Min1	Max1	Min2	Max2

2. The degree of adaptation of the new adjusted CA rules is calculated by validating whether the size in bits of the message to be hidden exceeds the capacity (selection threshold, selection threshold, etc.) of the message to be hidden. (k) of the message to be hidden exceeds the capacity (selection threshold,U) of the live pixels over time $t = 200$The adaptation margin will also depend on the evolution period of the GA in order to find the most suitable rule[4] . The adaptation margin will also depend on the percentage of live pixels at the end of the implementation in the CA, of the rule evolved after 25 generations. It has been noted through experimentation and literature consulted, that in order to achieve an efficient application of data

[4] During the testing of the GA with respect to the generations necessary to validate the resulting rule, it became evident that in generation 200 the transformation delta remained unchanged.

insertion by LSB (Least Significant Bit)-replacement in its variant established for this proposal, the percentage of selected or surviving pixels should not exceed 14% of the total pixels in an image. The following adaptation function has been defined for the implementation case:

$$f(x) = \begin{cases} 1, \quad si & \begin{cases} k_{LSB} \leq U_{P_V} \quad y \quad \dfrac{P_V}{P_T} * 100 \geq 0{,}12 \\ \\ k_{LSB} \leq U_{P_V} \quad y \quad \dfrac{P_V}{P_T} * 100 \leq 0{,}14 \end{cases} \\ & \quad \vdots \\ 0 & \quad new\ iteration \end{cases}$$

$$k\ is\ the\ size\ in\ bits\ of\ the\ message\ to\ hide,$$
$$U\ is\ the\ live\ pixel\ selection\ threshold\ for\ LSB - replacement\ (1,2,3), see\ Table\ 3$$
$$P_V\ is\ the\ total\ number\ of\ surviving\ pixels\ after\ the$$
$$evolved\ rule\ implementation$$
$$P_T\ is\ the\ total\ pixels\ of\ the\ treated\ image$$

3. For each iteration; a) a new rule is created by randomly deforming the genetic information, under the operations of Selection, Crossover and Mutation of binary strings [7][15] and adjusting positions according to minima and maxima, b) the degree of adaptation is validated, c) the rules are ordered according to the degree of adaptation, d) the rule with the highest degree of adaptation is selected and the resulting population is used for the data insertion process. A configuration of 4 initial individuals with a chromosome length of 8 bits and a mutation probability of 2% is used, the results are shown in Table 5:

Table 5. Initial population of 4 individuals.

A_1	0 0 0 0 0 0 1 0	2
A_2	0 0 0 0 0 0 1 1	3
A_3	0 0 0 0 0 0 1 1	3
A_4	0 0 0 0 0 0 1 1	3

- Selection process. After calculating the population size, the individuals are ordered, by their decimal value, from smallest to largest and half of the best population is selected [7]. See Table 6:

Table 6. Natural selection of the best population.

A_1	0 0 0 0 0 0 1 0	2	Progenitors
A_2	0 0 0 0 0 0 1 1	3	
A_3	0 0 0 0 0 0 1 1	3	
A_4	0 0 0 0 0 0 1 1	3	

- Crossover process. A crossover position is randomly generated along the chromosome of each individual, to subsequently join the head of the first chromosome with the tail of the second (A11, A22) and the head of the second with the tail of the first (A21, A12). Continuing with the example of the previous item, a random value between 1 and 7 is generated, as shown in Table 7:

Table 7. Reproductive cross in position 3.

A_1	0 0 0 0 0	0 1 0	2	0 0 0 0 0	0 1 1	3
	A11	A12		A11	A22	
A_2	0 0 0 0 0	0 1 1	3	0 0 0 0 0	0 1 0	2
	A21	A22		A21	A12	
	Crossing Point = 3, t=0			Crossing Point = 3, t=1		

Table 8. New population for the time $t = 1$reproductive crossbreeding.

A_1	0 0 0 0 0 0 1 0	2	Progenitors
A_2	0 0 0 0 0 0 1 1	3	
A_5	0 0 0 0 0 0 1 1	3	Descendants
A_6	0 0 0 0 0 0 1 0	2	

- Mutation process. In order to efficiently find a new rule for the evolutionary process of the CA, the Mutation procedure was implemented for the population resulting from the reproductive cross. It consists of randomly generating a position within the matrix formed by the four individuals, and modifying the value of the bit (gene) that occupies that position by its binary complement. If it finds in the position a '1', then it will be modified by a '0', in the same way, if it finds in the position a '0', then it will be modified by a '1'. For the case of implementation, it has been defined that the random value at the column level is generated in an interval between position 5 and 7, since, if we refer to the Moore neighborhood applied, the number of neighbors for each cell is a maximum of 8.

Table 9. New mutated population

	0	1	2	3	4	5	6	7		0	1	2	3	4	5	6	7	
0	0	0	0	0	0	0	1	0	2	0	0	0	0	0	0	1	0	2
1	0	0	0	0	0	0	1	1	3	0	0	0	0	0	0	1	1	3
2	0	0	0	0	0	0	1	1	3	0	0	0	0	0	0	0	1	1
3	0	0	0	0	0	0	1	0	2	0	0	0	0	0	0	1	0	2
	Pob (row, column)= (2,6), t=0									Pob (row, column)= (2,6), t=1								

At this point of the procedure, the evaluation of the resulting rule, R(2,3,1,2) is performed for $t = 25$ in the CA, in order to obtain a new population. The population of surviving pixels will be evaluated with the transition function, in order to verify the degree of adaptation achieved with the new rule, if it does not meet the maximum or minimum value required by the function, the entire GA procedure is applied again.

Continuing with the example raised from the development of the CA, the GA was executed during 200 iterations in order to optimize the evolution rule of the automaton, after reaching the maximum number of iterations in time in the GA and managing to evolve the CA during 25 generations for each iteration, the following data table was obtained. $t = 200$ in the GA and managing to evolve the CA during 25 generations for each iteration, the following data table was obtained, containing the rules that met the established adaptability criterion:

Table 10. Detail of the iterative process of the GA until the most suitable value is found**.**

Rule obtained after AG iteration	Iteration Number	Number of live pixels after application of AC	Percentage of Adaptability x >= 12% x <=14%
R(2,3,3,6)	64	325.739	12.358%
R(3,7,3,5)	68	342.518	12.995%
R(3,7,3,6)	71	364.193	13.817%
R(3,7,3,5)	88	342.518	12.995%
R(3,7,3,6)	97	364.193	13.817%
R(3,7,3,5)	124	342.518	12.995%
R(1,3,3,5)	**141**	**364.313**	**13.822%**

After analyzing the results, the implementation rule that optimizes the generational evolution of the CA, and therefore the number of surviving pixels within the allowed threshold, is obtained as R(1,3,3,5). Table 11 shows these values:

Table 11. Detail of available bits if the LSB replacement technique is applied.

Total pixels of treated image	Pixels for t=500	Available LSB-replacement bits (1)	Available LSB-replacement bits (2)	Available LSB-replacement bits (3)	
2.635.776	364.313 (13,8%)	1.092.939	2.185.878	3.278.817	bits
		136.617,38	273.234,75	409.852,13	bytes
		133,42	266,83	400,25	Kb
		0,13	0,26	0,39	Mb

Image 3 has been obtained as a result of the fusion between Cellular Automata and Genetic Algorithms, which elucidates in white color (live pixels), those pixels that managed to survive during the CA process and that will be used for the data insertion process.

5. Analysis of results of implementation variant "STEGANOS-AA".

In this phase of the process, with the clarity of pixels to be treated, the classical LSB procedure [12][13] focuses on modifying sequentially, from the last bit of each byte that makes up a pixel value (00011010, 11101011, 11111000), to the last three of each byte (00011010, 11101011, 11111000). It should be understood that the use of LSB has been strictly for reference for the availability analysis at each pixel value, it is not the technique implemented in the proposal. This modification exercise at each pixel value depends on the size of the message to be hidden and the level of impact on the resulting image.

The 'AA' variation, so called for this proposal, consists of marking in a position vector those locations, in the pixel values, that coincide with the byte strings of the information, so that the pixel itself does not suffer any modification by information to be embedded, conforming, through the referencing of ordered coordinates, the structure of a hidden message. This position vector is given by the implementation of a one-dimensional Cellular Automaton, whose operation is executed by means of Wolfram's rules 30 or 150 according to the case [1][8]. This method has been called in this work as *Modular Steganography*.

- *Implementation of the One-Dimensional Automaton:* According to Stephen Wolfram's notation [8], the specification of rules 30 and 150 are denoted by the following iterations. When a specific rule is applied to a row of initial states, a new row emerges in each iteration, which together form the matrix of generations, thus forming an image by applying a one-dimensional cellular automaton to an initial state. Figures 2 and 3 show the generation matrices obtained by means of rules R30 and R150 after applying 15 iterations [9].

Figure 2. Matrix of generations Rule 30. [9]

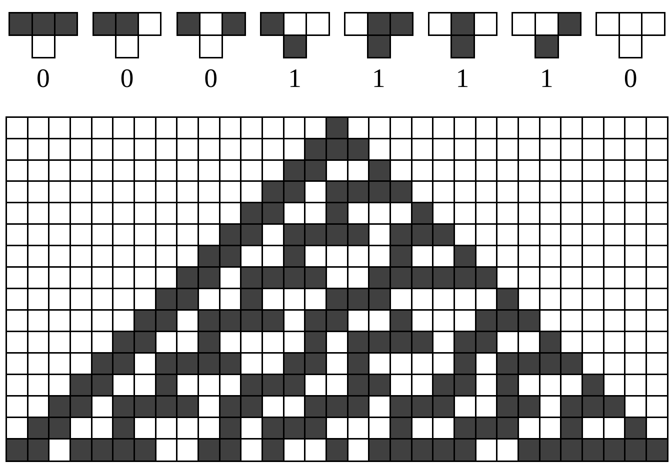
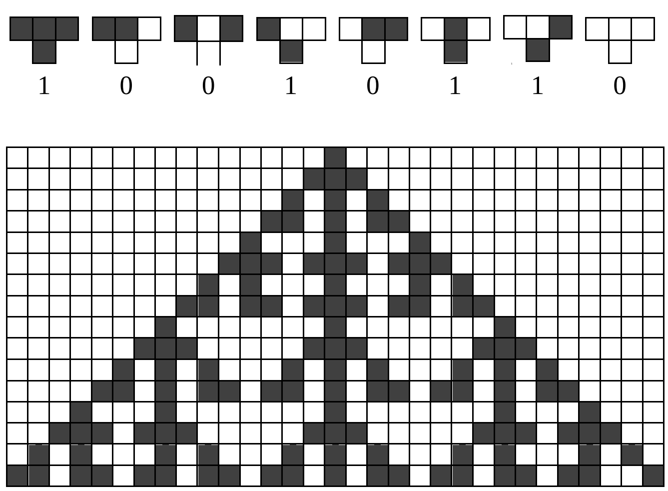


Figure 3. Matrix of generations Rule 150. [9]



Now we describe how this one-dimensional automaton interacts, under the implementation of rules 30 and 150, in the process of embedding or referencing by coordinates of the message to be hidden in the surviving pixels described in the previous items. The analysis exercise begins with the random selection of pixels in order to make it even more difficult to identify those on which the information to be hidden will be embedded (alternate LSB) or referenced (AA). The situation denotes the treatment in pixel number 324,732 (187, 197, 108) of the list obtained from the 363,656 surviving pixels, the one-dimensional automaton with Rule 30 is applied during 5 iterations[5] in each byte as initial state. The resulting bytes in the fifth generation will denote the new pixel value reference to be treated in the embedding process. This can be analyzed in Table 12:

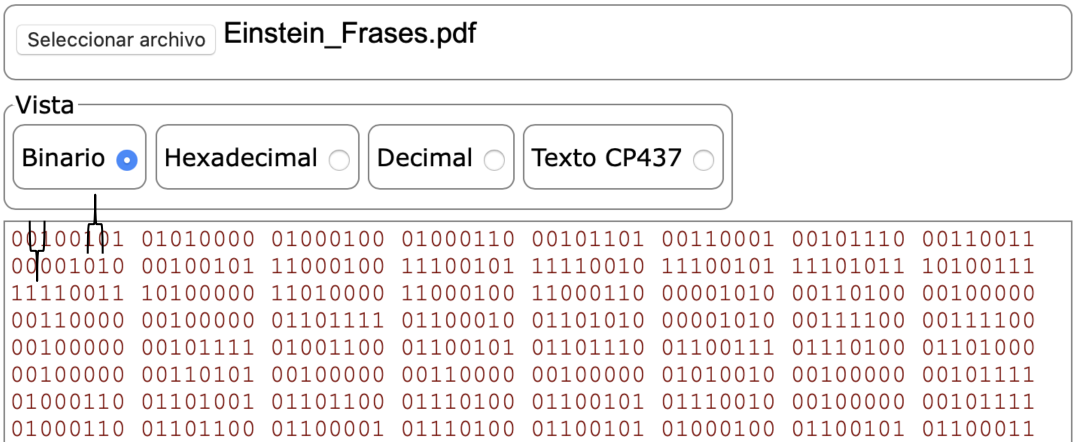
Table 12. Representation of generation matrix with R30.

g	R								G								B							
	187								197								108							
1	1	0	1	1	1	0	1	1	1	1	0	0	0	1	0	1	0	1	1	0	1	1	0	0
2	0	0	1	0	0	0	1	0	0	0	1	0	1	1	0	1	1	1	0	0	1	0	1	0
3	0	1	1	1	0	1	1	1	1	1	1	0	1	0	0	1	1	0	1	1	1	0	1	0
4	0	1	0	0	0	1	0	0	0	0	0	0	1	1	1	1	1	0	1	0	0	0	1	0
5	1	1	1	0	1	1	1	0	1	0	0	1	1	0	0	0	1	0	1	1	0	1	1	0

The information to be hidden, previously subjected to a binary conversion process, is sectioned into continuous tuples, composed of four bits each. Table 13 contains this data:

[5] As rows emerge for each iteration, the comparison values change. During testing, it was found that only the first five iterations allow the implementation of the referencing proposal.

Table 13. Binary representation of the file Einstein_Frases.pdf.



The first tuple to be arranged for embedding into the image, by means of ordered coordinate referencing, is $T_1 = \{0\ 0\ 1\ 0\}$. Now this tuple will be compared with the new pixel reference, obtained after the application of AC Rule 30. These results are shown in Table 14.

Table 14. Comparison of the 1era tuple of inf. to be hidden with respect to the pixel value, after 5G.

	INITIAL PIXEL BINARY									BINARY AFTER FIVE GENERATIONS RULE 30							
1	1	0	1	1	1	0	1	1		1	1	1	0	1	1	1	0
2	1	1	0	0	0	1	0	1		1	0	0	1	1	0	0	0
3	0	1	1	0	1	1	0	0		1	0	1	1	0	1	1	0

	BINARY AFTER 5G	POS1	POS2	POS3	POS4	POS5
1	$\{1\ 1\ 1\ 0\ 1\ 1\ 1\ 0\}$	$\{1\ 1\ 1\ 0\} \neq T_1$	$\{1\ 1\ 0\ 1\} \neq T_1$	$\{1\ 0\ 1\ 1\} \neq T_1$	$\{0\ 1\ 1\ 1\} \neq T_1$	$\{1\ 1\ 1\ 0\} \neq T_1$
2	$\{1\ 0\ 0\ 1\ 1\ 0\ 0\ 0\}$	$\{1\ 0\ 0\ 1\} \neq T_1$	$\{0\ 0\ 1\ 1\} \neq T_1$	$\{0\ 1\ 1\ 0\} \neq T_1$	$\{1\ 1\ 0\ 0\} \neq T_1$	$\{1\ 0\ 0\ 0\} \neq T_1$
3	$\{1\ 0\ 1\ 1\ 0\ 1\ 1\ 0\}$	$\{1\ 0\ 1\ 1\} \neq T_1$	$\{0\ 1\ 1\ 0\} \neq T_1$	$\{1\ 1\ 0\ 1\} \neq T_1$	$\{1\ 0\ 1\ 1\} \neq T_1$	$\{0\ 1\ 1\ 0\} \neq T_1$

After the sequential comparison of the tuple of information to be hidden and the representative values of the pixel, the expected merge was not achieved, this denotes that this pixel will not be used for hiding. Therefore, a new pixel should be selected and subjected to the same treatment with Rule 30. Table 15 shows the result:

Table 15. Generation matrix representation with R30 from 2[do] pixel.

g	R								G								B							
	75								64								44							
1	0	1	0	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	1	0	1	1	0	0
2	0	1	1	1	1	0	1	0	1	1	1	0	0	0	0	0	0	1	1	0	1	0	1	0
3	1	1	0	0	0	0	1	1	1	0	0	1	0	0	0	1	1	1	0	0	1	0	1	1
4	0	0	1	0	0	1	1	0	0	1	1	1	1	0	1	1	0	0	1	1	1	0	1	0
5	0	1	1	1	1	1	0	1	0	1	0	0	0	0	1	0	0	1	1	0	0	0	1	1

Table 16. Comparison of the 1[era] and 2[da] tuple of inf. to be hidden with respect to the pixel, after 5G.

	BINARY SECOND PIXEL								BINARY AFTER FIVE GENERATIONS RULE 30							
1	0	1	0	0	1	0	1	1	0	1	1	1	1	1	1	0
2	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0
3	0	0	1	0	1	1	0	0	0	1	1	0	0	0	1	1

	BINARY AFTER 5G	POS1	POS2	POS3	POS4	POS5
1	$\{0\,1\,1\,1\,1\,1\,1\,0\}$	$\{0\,1\,1\,1\} \neq T_1$	$\{1\,1\,1\,1\} \neq T_1$	$\{1\,1\,1\,1\} \neq T_1$	$\{1\,1\,1\,1\} \neq T_1$	$\{1\,1\,1\,0\} \neq T_1$
2	$\{0\,1\,0\,0\,0\,0\,1\,0\}$	$\{0\,1\,0\,0\} \neq T_1$	$\{1\,0\,0\,0\} \neq T_1$	$\{0\,0\,0\,0\} \neq T_1$	$\{0\,0\,0\,1\} \neq T_1$	$\{0\,0\,1\,0\} = T_1$
3	$\{0\,1\,1\,0\,0\,0\,1\,1\}$	$\{1\,0\,1\,1\} \neq T_2$	$\{0\,1\,1\,0\} \neq T_2$	$\{1\,1\,0\,1\} \neq T_2$	$\{1\,0\,1\,1\} \neq T_2$	$\{0\,1\,1\,0\} \neq T_2$

After a new sequential comparison of the tuple of information to be hidden and the representative values of the pixel, the expected merge is achieved in the second byte of the second selected pixel, the pixel 134,968 (75, 64, 44) of the list of survivors, this denotes that this pixel will be demarcated within a matrix that conforms the password module for the desteganography process. Then the first reference coordinate, for the first tuple of the secret message, will be: $Cr_0 = \{30{,}134968{,}1{,}0{,}3{,}5\}$. Table 16 shows that the third byte of the analyzed pixel was not compared with the first tuple of the secret message. T_1 since that tuple was referenced in the previous byte, therefore, the new tuple $T_2 = \{0\ 1\ 0\ 1\}$ is subjected to a new comparison process for its respective referencing.

In this way, we continue with the referencing by coordinates of the information to be hidden by means of the implementation of the one-dimensional AC of Wolfram[6] . Finally, a password module is obtained with the complete reference coordinates for each byte that makes up the secret message and a stegoimage identical to the original one, without any type of modification, therefore, INDETECTABLE BY THE RESEARCHED STEGOANALYSIS ALGORITHMS.

The hiding exercise using the 'AA' variant, in its native version, does not involve any modification of the original image, therefore, the test routine generated positive results against the suspicion of hidden information. The following attacks [6][17] were applied, with 100% optimal results, against the implementation of the proposed technique, none of them detected suspicious information: Visual Attack, Chi-square Attack and RS Attack.

6. Conclusions

During the process of implementation of the native action of cellular automata, the mathematical and predictive context of the combination of rules in two (2d) dimensions, with the Moore neighborhood, and in one (1d) dimension by applying Stephen Wolfram's rules 30 and 150, was verified. The unusual prediction of a suitable population of pixels and the logical treatment of their values for the purpose of data and information hiding, resulted in an interesting and indecipherable method that allows corroborating the theoretical effect described by Dr.

[6] Dr. Stephen Wolfram participated in a virtual conference called "A New Fundamental Theory of Physics" on June 6, 2020, as part of a series of conferences organized by the Astrophysics Institute of the Pontificia Universidad Católica de Chile, in this conference, in which we participated as assistants, Dr. Wolfram presented his latest research that continues postulating Cellular Automata as inverse methods that result in extraordinary instruments when combined with computational sciences. Indeed, this confirmation in front of world-class scientists boosted the object of the research by supporting with theoretical and practical arguments the proposal described in detail in that paper.

Wolfram, for the case of automata implemented in only one dimension, through the obtaining of complex systems by means of simple models.

The causal connections determined in the fusion of Genetic Algorithms and Cellular Automata resulted in a new steganographic method that through the simple interaction of generational formulas manages to demonstrate complex levels of random selection and a modular analysis of referenced occultation.

The level of interaction reached with the implementation of Genetic Algorithms (GA), not only denoted an interesting and novel case of implementation of this artificial intelligence method, but also proved the efficiency of GA's as optimization methods. During the research, the intervention of GA's reached its greatest participation when improving the selection rules, coming from the evolutionary rules of the Cellular Automaton (CA) proposed as initial solution. As the GA optimizes the selection rules, the CA, in an optimized way, reaches its maximum level of evolution, resulting in obtaining suitable rules for the selection of candidate pixels for the hiding process.

Through the research process, several of the existing techniques to execute the data hiding process were analyzed [12], covering several interesting methods of implementation. These techniques offered the possibility of proposing a new technique based on the initial objects of study of this project, resulting in a novel Steganographic approach based on GAs and CAs. MODULAR STEGANOGRAPHY, AS THIS NEW TECHNIQUE HAS BEEN DEFINED, DENOTES A NEW KNOWLEDGE RESULTING FROM THE RESEARCH PROCESS PRESENT IN THIS PROJECT.

Modular Steganography consists of a steganographic technique that through a generational analysis of each pixel value, subjected to a strict procedure based on CA's and GA's, allows data hiding through the referencing of coordinates ordered and structured by a One-Dimensional Cellular Automaton, obtaining an image identical to the original one and a password file module, necessary for the inverse hiding process. While Steganography focuses on data hiding through image processing, this new postulate proposes to hide data by referencing its content in the pixel values resulting from the CA intervention.

The technique presented offers an asymmetric scenario of concealment, since if the information sent (treated image and password module) is intercepted by different channels and at different times, the process of de-steganography becomes more complex, since, as evidenced in the process of image analysis, no suspicion of concealment arises, and in the case of the password module, the process of reviewing a "supposed" concealment denotes another type of in-depth analysis.

As an alternative for masking and using the variant of the LSB technique described for the selection of suitable pixels in the process of defining the image capacity, it is possible to propose a nested technique that modifies the carrier image by a minimum percentage, through the random selection of pixels resulting from the SteganosAA treatment.

References

[1] S. Wolfram, and N. H. Packard, "Two-dimensional Cellular Automata," J. Statist. Phys., vol. 38, 1985. https://doi.org/10.1007/BF01010423

[2] J. D. Muñoz Castaño, "Autómatas Celulares y Física Digital," in Memorias del Primer Congreso Colombiano de Neuro Computación, Santa fe de Bogotá, D.C.: Academia Colombiana de Ciencias Exactas, Físicas y Naturales, 1996, p. 28.

[3] J. C. Hernández, "Algunas Generalizaciones en Autómatas Celulares," Consejo Nacional de Ciencia y Tecnología - CONACYT, Mexico, 2008.

[4] G. Juárez, "Teoría del Campo Promedio En Autómatas Celulares Similares a 'The Game Of Life'," Centro de Investigación y Estudios Avanzados del Instituto Politécnico Nacional, Mexico, 2000.

[5] E. Cuevas, D. Zaldívar, and M. Pérez, "Procesamiento digital de imágenes con MATLAB y Simulink," Alfaomega Grupo Editor; RA-MA Editorial, Mexico, 2010.

[6] M. A. Muñoz, "Privacidad y ocultación de información digital: Esteganografía protegiendo y atacando redes informáticas," Ra-ma, Ediciones de la U, Madrid, Bogotá, Spain, Colombia, 2017.

[7] C. Ponce, "Inteligencia Artificial con aplicaciones a la ingeniería," Alfa Omega Grupo Editor, Mexico, 2010.

[8] W. S. Wolfram, "Cellular automata as simple self-organizing systems," Caltech preprint CAL-68-938, Pasadena, 1982.

[9] M. Espínola, "Clasificación de Imágenes de Satélite mediante Autómatas Celulares," Universidad de Almería, Almería, 2011.

[10] E. F. Moore, "Machine Models Of Self-Reproduction," in Proceedings of Symposia in Applied Mathematics, vol. 14, 1963. https://doi.org/10.1090/psapm/014/9961

[11] C. Á. Guerrero, "RapaNui - Isla de Pascua," RapaNui, Chile, /2018.

[12] A. Cheddad, J. Condell, K. Curran, and P. McKeitt, "Digital image steganography: Survey and analysis of current methods," School of Computing and Intelligent Systems, University of Ulster at Magee, Northern Ireland, Signal Processing, vol. 90, no. 3, 2010. https://doi.org/10.1016/j.sigpro.2009.08.010

[13] A. De La Cruz Franco, "Implementación de un Algoritmo Computacional para Esteganografía basado en técnicas del bit menos significativo," Universidad de Quintana Roo, Chetumal, Mexico, 2017.

[14] J. I. Vázquez and J. Oliver, "Evolución de Autómatas Celulares utilizando Algoritmos Genéticos," Universidad de Deusto, Bilbao, Spain, 2008.

[15] A. Miri and K. Faez, "Adaptive Image Steganography based on transform domain via Genetic Algorithm," Department of Electrical Engineering, Amirkabir University of Technology, Tehran, Iran, Optika, vol. 145, 2017. https://doi.org/10.1016/j.ijleo.2017.07.043

[16] S. Mukjerjee, S. Roy, and G. Sanyal, "Image Steganography Using Mid Position Value Technique," National Institute of Technology Durgapur, Durgapur, India, Procedia Computer Science, vol. 132, 2018. https://doi.org/10.1016/j.procs.2018.05.160

[17] A. Westfeld and A. Pfizmann, "Attacks on Steganographic Systems," Department of Computer Science, Dresden University of Technology, Dresden, Germany, Information Hiding, 1999. https://doi.org/10.1007/10719724_5

[18] H. Caballero, "Cálculo de la dispersión de pixels en imágenes RGB para Esteganografía con base en la teoría fractal," Facultad de Ingeniería, Universidad Autónoma de México, Toluca de Lerdo, Mexico, 2020.

[19] J. Fridrich, M. Goljan, and R. Du, "Reliable Detection of LSB steganography in color and grayscale images," Department of Electrical and Computer Engineering, Binghamton University, Binghamton, U.S.A., 2002. https://doi.org/10.1145/1232454.1232466