COMPRESION DE IMAGENES: UN ENFOQUE DE AUTOMATAS CELULARES EVOLUTIVOS

HILDEMAR JOSE MARTINEZ DIAZ

UNIVERSIDAD CENTROCCIDENTAL "LISANDRO ALVARADO"

BARQUISIMETO, 2000

COMPRESION DE IMAGENES: UN ENFOQUE DE AUTOMATAS CELULARES EVOLUTIVOS

Por

Hildemar José Martínez Díaz

Proyecto de Tesis de Grado para optar

al grado de Magister Scientiarum

UNIVERSIDAD CENTROCCIDENTAL "LISANDRO ALVARADO"

Decanato de Ciencias y Tecnología

BARQUISIMETO, 2000

UNIVERSIDAD CENTROCCIDENTAL
"LISANDRO ALVARADO"
DECANATO DE CIENCIAS Y TECNOLOGIA
COORDINACION DE POSTGRADO

ACTA DE VEREDICTO DE TRABAJO DE GRADO

Nosotros, Miembros del Jurado Examinador del Trabajo de Grado titulado: "*COMPRESION DE IMÁGENES: UN ENFOQUE DE AUTOMATAS CELULARES EVOLUTIVO*", **presentado por el Ingº MARTINEZ DIAZ, Hildemar José,** titular de la Cédula de Identidad Nº 9.620.373, como requisito para optar al grado académico de **MAGISTER SCIENTIARIUM EN INTELIGENCIA ARTIFICIAL,** ofrecido por el programa de Maestría en Inteligencia Artificial del Decanato de Ciencias y Tecnología de la Universidad Centroccidental "Lisandro Alvarado", hacemos constar que hoy, catorce de Abril del dos mil (14-04-2000) a las tres de la tarde (3:00 p.m.) se realizó el examen Público de Defensa de Trabajo de Grado, de acuerdo a lo establecido en la Normativa sobre Trabajos de Grado de la UCLA. Una vez rendido el examen, este Jurado emite el siguiente veredicto: El Trabajo de Grado fue: **APROBADO CON MENCION HONORIFICA Y PUBLICACION.**

Dando fé de ello, levantamos la presente acta en la ciudad de Barquisimeto a los catorce días del mes de Abril del dos mil.

PROF. ALBERTO CASTILLO VICCI
Presidente del Jurado
C.I.Nº 1.277.995

PROF. JOSE ALI MORENO
Jurado Principal
C.I.Nº 3.203.084

PROF. MARITZA BRACHO DE RODRIGUEZ
Jurado Principal
C.I.Nº 4.490.568

Dedico esta tesis a todas aquellas personas que de una u otra manera creen en la importancia que la ciencia tiene en nuestra vida y la necesidad que hay de desarrollarla.

AGRADECIMIENTO

A mis padres, quienes supieron inculcarme en todo momento la importancia del estudio y la superación personal ante cualquier eventualidad, de las que diariamente conseguimos en nuestro camino.

A mi tutor José Alí Moreno, por enseñarme a apreciar los Autómatas Celulares, Algoritmos Genéticos y los métodos de Computación emergente en general; y por mostrarme como se puede combinar la ciencia con lo cotidiano.

A Maritza Bracho, por su interés en este trabajo y el apoyo y consejo oportuno que en todo momento me supo dar.

A mis compañeros de maestría, aquellos que no pudieron continuar y a quienes recuerdo y sobre todo a los que finalmente conseguimos esta meta luego de pacientemente sortear innumerables obstáculos; al igual que ellos, creí y desarrollé una fe inquebrantable de finalizar las metas propuestas.

CURRICULUM VITAE

Hildemar José Martínez Díaz
Candidato para obtener el grado de Magister Scientiarum
Tesis: Compresión de Imágenes: Un enfoque de Autómatas Celulares Evolutivos
Posgrado: Inteligencia Artificial

Ingeniero en Informática, graduado en la Universidad Centroccidental "Lisandro Alvarado" desde el año 1993. Ha trabajado en la industria petrolera y en diferentes tipos de empresas como consultor en la implantación de diversos productos computacionales. Posee tres publicaciones internacionales relacionadas con el tema de esta tesis.

Actualmente trabaja como Gerente en una firma internacional en el área de Auditoría de Sistemas, servicios de análisis de riesgos tecnológicos y seguridad en plataformas computacionales.

COMPRESION DE IMAGENES: UN ENFOQUE DE AUTOMATAS CELULARES EVOLUTIVOS

Hildemar José Martínez Díaz

RESUMEN

En este trabajo se presenta una técnica para comprimir imágenes usando Autómatas Celulares Evolutivos (ECA) binarios para comprimir imágenes. El método es aplicado a imágenes de huellas digitales en blanco y negro. Estos Autómatas Celulares Evolutivos son evolucionados con un Algoritmo Genético y guiados por una función de adaptación que consiste en una medida de similitud entre la configuración de los Autómatas Celulares y la imagen objetivo. Cuando se consigue una aproximación adecuada de la imagen objetivo, esta puede ser codificada con números de reglas y las operaciones lógicas que encadenan dichos Autómatas Celulares. De esta manera, la imagen original sufre de un proceso de compresión, ya que la misma puede ser representada con una cantidad considerablemente menor de bits. Con la experimentación, se obtienen ratas de compresión por el orden de 700:1. Una imagen codificada o comprimida puede ser regenerada, recuperando una buena aproximación de la imagen original al poseer una similitud del 96%. El método es comparado en velocidad y ratas de compresión con otros algoritmos de compresión ampliamente comercializados.

Palabras claves: Autómatas Celulares, Algoritmos Genéticos, Compresión, Procesamiento de Imágenes.

IMAGE COMPRESSION: AN EVOLVED CELLULAR AUTOMATA FOCUS

Hildemar José Martínez Díaz

ABSTRACT

In this work a technique for image compression using binary Evolutionary Cellular Automata (ECA) is presented. The method is applied to black and white fingerprint images. These ECA are evolved with a Genetic Algorithm (GA) guided by a fitness function consisting of a similarity measure between the configurations of the ECA and the target image. When a suitable approximation of the target image is achieved it can be codified with the rule numbers and logical operations linking the ECA. In this way the original image undergoes a compression process since it will be represented by a considerably smaller bit amount. Typically compression rates of the order of 100:1 are achieved in the experiments. A codified or compressed image can be regenerated recovering a very good approximation with a typical similarity of 96% to the original image. The method is compared in speed and compression rate with other commercially and widely used compression algorithms.

Keywords: Cellular Automata, Genetic Algorithms, Compression, Image Processing.

I N D I C E

INDICE DE FIGURAS

I. INTRODUCCION

A. Introducción General

Cuando una ser humano se encuentra frente a un problema de gran complejidad, la primera reacción que se tiene es comenzar a resolverlo con las herramientas disponibles, sin a veces siquiera pensar en ir a los niveles más bajos de detalles, para simplificarlo y poder llegar a una solución satisfactoria de la manera más adecuada. Sin embargo, cuando esas herramientas fallan, es cuando se inicia el proceso de análisis profundo que debió ser efectuado en un principio, para poder obtener resultados.

Comportamientos como este último, son similares a las desarrolladas por técnicas recientes de resolución de problemas. En estas se utiliza el poder de cálculo del computador y la manera inteligente de comportarse de los sistemas naturales. De allí surge la rama de la ciencia de la computación conocida como Computación Emergente. Estas técnicas se basan en imitar en las computadoras el comportamiento humano o de los sistemas naturales. Para ello, se han creado modelos de razonamiento, aprendizaje y algunas estrategias relacionadas con los mismos. De esta manera, se han imitado por ejemplo el comportamiento de las neuronas humanas y sus procesos sinápticos, lo cual ha dado la razón de ser de las Redes Neuronales.

Dentro de este mismo orden, se han creado los Algoritmos Genéticos que representan el paradigma de la evolución natural desde el punto de vista computacional. Este término apareció por primera vez en 1967, aún cuando es en 1975 que John Holland en su libro Adaptación de Sistemas Naturales y Artificiales crea una corriente de investigación sobre

los mismos. En la actualidad, los Algoritmos Genéticos están siendo ampliamente utilizados en aplicaciones robóticas, para el aprendizaje de Redes Neuronales y en el reconocimiento de patrones.

También parte de estos estudios de Computación Emergente incluyen a los Autómatas Celulares, los cuales son sistemas dinámicos que se caracterizan por el hecho de que el espacio y el tiempo son discretos. Los Autómatas Celulares fueron introducidos por primera vez por John von Neumann, para modelar los sistemas auto-reproductores.

Por otro lado, es bien conocido que los Autómatas Celulares son capaces de generar una amplia variedad de patrones, comenzando desde un estado inicial de un conjunto de células e iterando mediante el uso de sus reglas dinámicas. Este comportamiento será la base para el desarrollo de este trabajo, que permite representar una imagen gráfica como un conjunto de autómatas y operadores booleanos.

B. Definición del Problema

La idea principal de este trabajo, es poder diseñar un algoritmo de compresión con el uso de Autómatas Celulares y Algoritmos Genéticos.

Es bien conocido que los Autómatas Celulares uni-dimensionales poseen la cualidad de generar formas bastante definidas al ser aplicadas sobre un estado inicial de células y tomando en cuenta el rango de interacción entre las mismas. Estos pueden generar una cantidad a veces muy amplia de formas que son estrictamente dependientes del estado

inicial y la regla de evolución aplicada a los mismos. Haciendo uso de esta característica, se pretende diseñar un algoritmo que permita localizar de la manera más efectiva posible un conjunto de reglas de Autómatas Celulares y operadores los cuales entre sí puedan recrear la imagen objeto de compresión.

C. Objetivo General

Diseñar un algoritmo para la compresión y descompresión de imágenes haciendo uso de Autómatas Celulares Evolutivos, el cual pueda ser aplicable a imágenes gráficas de computadoras.

D. Objetivos Específicos

a. Diseñar el algoritmo de manera general, para que pueda ser utilizado en cualquier aplicación bien sea de software o hardware.

b. Producir un algoritmo que permita de manera eficiente efectuar la compresión y descompresión de imágenes en un tiempo adecuado.

c. Hacer el análisis de los parámetros del Algoritmo Genético que son necesarios, para obtener la mejor ejecución del mismo al efectuar la compresión.

d. Crear un esquema inicial, que pueda ser ampliado, para extender el algoritmo de compresión a otras fuentes de datos.

E. Justificación

La cantidad de información que es necesaria y que esta disponible se incrementa cada vez más, por ello se necesitan formas muy eficientes de representar esta información. La compresión de imágenes y de datos en general se ha convertido en un objetivo primordial a lograr, ya que la búsqueda de la reducción del número de bits usados para mantener una misma información hace que sea más barato su almacenamiento o transmisión. Existe una amplia cantidad de soluciones basadas en Hardware y Software para lograr una efectiva forma de comprimir información, las cuales raras veces tienen algo en común, salvo el hecho de efectuar la compresión de la información.

El objetivo principal de la compresión, es de representar la información de la manera más eficiente posible. Esto es posible lograrlo, con técnicas que usan los diferentes tipos de estructuras que están presentes dentro de dicha información. La información puede tener múltiples formas como texto, imágenes, sonidos, voz, video y otras. Cada una de estas formas posee su propia estructura interna, aunque también poseen características comunes que permiten el diseñar mecanismos de compresión que sean iguales para todos los tipos de información.

Es por ello que la búsqueda de nuevas y diferentes maneras de comprimir información es necesaria, para lograr más rápidas y mejores maneras de optimizar los espacios de almacenamiento, así como su posterior transmisión.

F. Alcances y Limitaciones

1. Alcances

Cuando se hace el desarrollo de nuevos algoritmos y se comparan contra los algoritmos más conocidos que efectúan la misma tarea, es importante el tomar un conjunto de datos que establezcan un entorno común de pruebas comparativas. De allí que la información de entrada del algoritmo desarrollado será una imagen de 256 por 256 bits de tamaño, la cual es considerada como suficientemente representativa para realizar pruebas sobre algoritmos de compresión de imágenes gráficas. A partir de esta información, se generan poblaciones de Autómatas Celulares, los cuales evolucionaran para recomponer la imagen original.

2. Limitaciones

Este algoritmo desarrollado, es apenas una primera aproximación de lo que se puede lograr usando este mecanismo, por lo tanto se usarán sólo imágenes de prueba en dos tonos, blanco y negro. No obstante, es posible el desarrollar y extender el mismo concepto aquí presentado para lograr compresiones de imágenes a color con ciertas modificaciones al algoritmo básico.

Para las pruebas, se utilizaron imágenes que representan impresiones digitales. Sin embargo, también puede extenderse su uso a cualquier imagen en blanco y negro.

II. MARCO TEORICO DE LA INVESTIGACION

En este trabajo, se discute la combinación de diferentes tipos de conocimientos de la Inteligencia Artificial, para producir una sinergia en la posibilidad de conseguir una nueva manera, utilizando métodos evolutivos, para realizar la compresión de imágenes en blanco y negro y con un porcentaje adecuado de compresión.

Para ello se usarán los Autómatas Celulares y los Algoritmos Genéticos en un entorno conocido como Autómatas Celulares Evolutivos, en la búsqueda de una posibilidad de realizar dicha compresión. A continuación se da una descripción de los conceptos aplicados en el desarrollo de este trabajo.

A. Autómatas Celulares

1. Concepto de Autómatas Celulares

Los Autómatas Celulares fueron introducidos por primera vez en 1948 de manera independiente por John von Neumann (1966) y S. M. Ulam, con la finalidad de proveer un marco formal para la investigación del comportamiento de sistemas complejos. Un Autómata Celular es definido como un sistema dinámico, donde el tiempo, el espacio y el estado del sistema en general se caracterizan por ser discretos.

Los Autómatas Celulares han sido estudiados como objetos matemáticos, como modelos de sistemas naturales y como arquitectura para un rápido y seguro desarrollo de sistemas altamente paralelos, Wolfram (1984).

Un Autómata Celular es un reticulado de espacio regular donde cada punto es llamado celda, dicha celda puede poseer alguno de un número finito de estados definidos en el sistema. Este estado es actualizado dependiendo de una regla dinámica o función de transición y del estado exactamente anterior de la celda y las celdas que la rodean, Wolfram (1984). Todos los estados de las celdas en el reticulado son actualizados de manera sincrónica. De esta manera, el estado de todo el reticulado es definido en tiempos discretos.

El arreglo de celdas de un Autómata Celular puede ser n-dimensional, en donde en mayor medida se emplean los valores de una, dos o tres dimensiones. La regla contenida en cada una de las celdas es, en esencia, una máquina de estado finito, la cual está definida en una tabla de reglas o función de transición, que representa una relación de las posibles configuraciones de los estados de las celdas.

La configuración de un Autómata Celular uni-dimensional con estados binarios (0,1), puede ser representada como en la siguiente figura:

Figura 1. Autómata Celular Uni-dimensional

1	0	1	0	0	1	1	1	0	1	0

Cuando se utilizan configuraciones de un número finito de celdas, es común el tomar condiciones de frontera, lo cual resulta en una rejilla circular en el caso de Autómatas Celulares uni-dimensionales y un toroide para el caso bi-dimensional.

2. Función de Transición de los Autómatas Celulares

La regla dinámica para la evolución de los autómatas celulares, para el caso unidimensional, puede expresarse como la siguiente función de transición de estados:

$$S_i(t + 1) = F[S_{i-r}(t), S_{i-r+1}(t),..., S_i(t),..., S_{i+r-1}(t), S_{i+r}(t)]$$

donde r es el rango de intersección (o también llamado radio) entre las células, F es la función de transición y $S_i(t)$ representa el estado de la celda i en el tiempo t.

El conjunto de células (i-r,...,i,...i+r) cercanas a la célula i, se define como la vecindad de i y cuya notación es N_i. Cuando las células de una vecindad toman valor, entonces se dice que se ha definido una configuración para esa vecindad.

La dinámica de los Autómatas Celulares, es una relación del conjunto de todas las posibles configuraciones de las vecindades N, al conjunto de todos los estados S de la celda i. De allí, que al haber k estados posibles, habrá entonces k^{2r+1} configuraciones de vecindad posibles, permitiendo definir la relación:

$$N \rightarrow S \ (S \in N, N \in Z^{2r+1}),$$

donde Z representa el valor k de estados posibles.

Para la representación de la función de transición en el caso binario, se utiliza una tabla de reglas como en el siguiente ejemplo para r = 1:

Figura 2. Tabla de Reglas del Autómata

Vecindad (N)	000	001	010	011	100	101	110	111
Estado (S)	0	0	0	1	0	1	1	1

Como puede observarse, dada que las posibilidades de vecindades son fijas, la función de transición de estados para el caso presentado en la Figura 2, puede representarse con un número entero, el cual no es más que la transformación de los estados resultantes de la regla presentada anteriormente como un número binario. De esta manera, la regla puede expresarse como 11101000, o con el número entero 232, tomando en cuenta que el primer estado resultante de la primera configuración de la vecindad es el bit menos significativo del número a construir.

Más formalmente, esto puede expresarse usando la siguiente tabla de vecindades y estados:

Figura 3. Tabla de Vecindades y Estados del Autómata

Vecindad (N)	000	001	010	011	100	101	110	111
Estado (S)	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7

De allí se tiene que la función de transición en este caso (r = 1 y estados binarios), se puede expresar mediante el número calculado con la fórmula:

$$\text{Regla} = c_0 \times 2^0 + c_1 \times 2^1 + c_2 \times 2^2 + c_3 \times 2^3 + c_4 \times 2^4 + c_5 \times 2^5 + c_6 \times 2^6 + c_7 \times 2^7$$

Al aplicar la regla anteriormente presentada al Autómata Celular uni-dimensional de la Figura 1, se obtendría la siguiente configuración:

Figura 4. Autómata Celular luego de la aplicación de la Regla del Transición

0	1	0	0	0	1	1	1	1	0	1

Estas mismas consideraciones pueden ser extendidas para el caso de vecindades de dos y tres celdas (r = 2 ó r = 3).

3. Autómatas Celulares Legales

Para poder limitar el número de Autómatas Celulares que sean manejables y aún tener la posibilidad de definir sistemas naturales, se ha definido el concepto de Autómata Celular Legal.

Por definición, un Autómata Celular Legal es aquel que cumple con las siguientes condiciones:

a) Condición base: Existen algunas configuraciones las cuales permanecen sin variación. De esta forma {Si} = {0} y F(0,..., 0) = 0. Debido a esta condición en la función de transición, una vez que una celda de un Autómata Celular ha tomado un estado con valor de cero, mantiene su estado invariable.

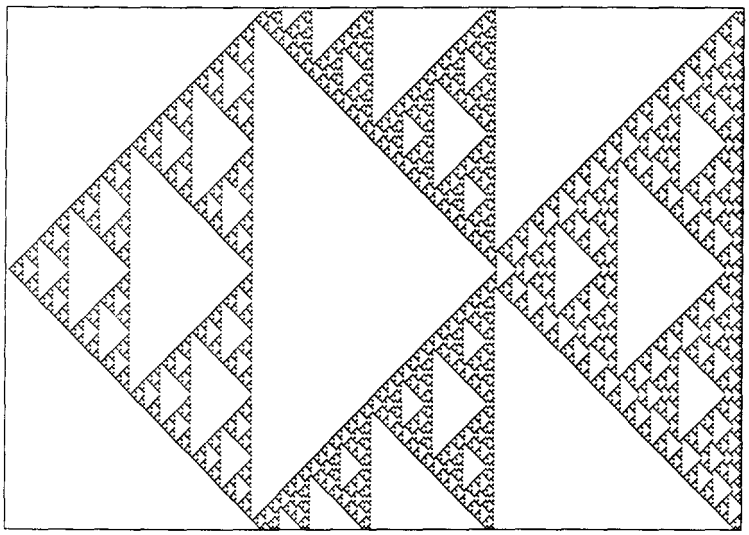
b) En general, la regla de transición es local pero homogénea; es decir, no depende de la posición de i de la celda en el reticulado. Por tanto, se dice que posee dinámica homogénea y frecuentemente se asume también que esta función de transición es simétrica, es decir que $F(.....,s_{i-j},....,s_{i+j},.....) = F(.....,s_{i+j},....,s_{i-j},.....)$. Dos ejemplos de esta condición pueden ser: $F(11000) = F(0001\,1)$ *ó* $F(10010) = F(01001)$.

Estas dos condiciones reducen el número de transiciones posibles de $k^{k^{2r+1}}$ a $k^{k^{r+1}(k^r+1)/2-1}$. Los Autómatas Celulares que poseen las dos condiciones anteriores, si bien es cierto que se reducen significativamente en número, aún representan una gran cantidad inclusive para valores de k y r pequeños.

4. Ejemplo de Autómata Celular

La siguiente figura de ejemplo es producto de la evolución de un Autómata Celular binario, uni-dimensional, finito, con trescientas celdas, vecindad de una sola celda (r = 1) y regla de evolución noventa (90). El reticulado se inicializa con solo 1 bit en el centro del mismo.

Figura 5. Ejemplo de Autómata Celular



B. Algoritmos Genéticos

1. Los Algoritmos Genéticos y su Analogía con la Evolución

La evolución de los seres vivos y sus características fueron de alta motivación para que John Holland, en la década iniciada en 1980, comenzara una línea de investigación en esta área, lo cual posteriormente se transformó en los Algoritmos Genéticos que se conocen hoy en día. Los algoritmos que en un comienzo fueron desarrollados por Holland, a pesar de ser sencillos, dieron excelentes resultados en la solución de problemas considerados para la época como difíciles.

Estos algoritmos se basan en las características de la evolución, las cuales fueron descritas por Lawrence Davis (1991) de la manera siguiente:

a) La evolución es un proceso que opera en los cromosomas, en lugar de operar en los seres vivos que ellos codifican.

b) Los procesos de selección natural provocan que aquellos cromosomas que codifican estructuras con éxito, se reproduzcan más frecuentemente que aquellos que no poseen estas estructuras.

c) Las mutaciones pueden causar que los cromosomas de los hijos sean diferentes a los de los padres y los procesos de recombinación pueden crear cromosomas bastante diferentes en los hijos por la combinación del material genético de los cromosomas de los dos padres.

d) La evolución biológica no tiene memoria.

La premisa de todo Algoritmo Genético es que se pueden encontrar soluciones aproximadas a problemas de gran complejidad computacional, mediante un proceso de evolución simulada utilizando un algoritmo matemático en una computadora. Al igual como ocurre en la evolución biológica, en la evolución simulada se hará una búsqueda ciega de la solución mediante la manipulación de la información intrínseca a dichas soluciones. Un Algoritmo Genético puede ser visto entonces, como una estructura de control que organiza o dirige un conjunto de transformaciones y operaciones diseñadas para simular los procesos de evolución.

En la evolución de los seres vivos, la supervivencia es el problema a lo que cotidianamente se enfrenta cada individuo. Para ello cada uno de ellos cuenta con ciertas características innatas en su material genético para hacerle frente a dichos problemas. A nivel de los genes, el problema es buscar aquellas adaptaciones beneficiosas para el medio ambiente hostil y cambiante en extremo. De allí que gracias al mecanismo de selección natural, cada individuo adquiere cierto conocimiento que es codificado y agregado a sus genes. Estos son a su vez modificados por las operaciones de reproducción. Algunas de ellas son mutaciones aleatorias, inversión de partes de los cromosomas y el entrecruzamiento, el cual es definido como el intercambio de material genético por parte de los dos padres.

En los Algoritmos Genéticos las mutaciones aleatorias proveen cierta variación y ocasionalmente, introducen cambios beneficiosos en los cromosomas. El entrecruzamiento por su parte, es el mecanismo que altera la ubicación de los genes, permitiendo el intercambio de material genético entre los dos padres para crear estructuras hijas.

2. Comparación contra los Métodos Tradicionales de Búsqueda

a. Tipos de Métodos de Búsqueda.

En la actualidad se conocen tres tipos de métodos de búsqueda: los métodos basados en cálculo, los enumerativos y los aleatorios.

Los métodos basados en cálculo se dividen a su vez en dos tipos: los directos y los indirectos. Los indirectos, buscan una solución local al problema haciendo uso de un

conjunto de ecuaciones no lineales, las cuales se deben resolver, haciendo el gradiente de la función objetivo igual a cero. Gráficamente este método encuentra los picos de una función restringiendo la búsqueda a todos los puntos que tengan inclinación cero en todas las direcciones. Por otra parte los métodos directos, buscan los picos locales haciendo un salto en la función y moviéndose en una dirección relacionada al gradiente local. Este es el mecanismo que es utilizado en el "hill-climbing" o en el descenso de gradiente. Estos métodos han sido mejorados de muchas maneras y poseen el mismo inconveniente de que carecen de robustez. Ambos métodos tienen un ámbito local, es decir que el optimo es localizado entre los mejores puntos que están cercanos al punto actual. De allí que el comenzar cerca de un pico pequeño haría que se consiguiera como solución ese punto y tal vez no alguno que se encuentre un poco más alejado y que sea el pico más alto de la función. De allí que las mejoras involucran un componente aleatorio para efectuar un reinicio que permita localizar los picos más altos. Otro problema con los métodos basados en cálculo es también que ellos son útiles solo en el caso de que exista la derivada de la función objetivo y esta no siempre existe para todos los problemas de búsqueda.

Los métodos enumerativos por su parte, han sido considerados de muchas maneras y formas. Su funcionamiento es sencillo y se basa en definir un espacio de búsqueda finito a partir de un espacio de búsqueda infinito y discretizado, e ir evaluando la función objetivo en cada uno de los puntos del espacio previamente definido. Por supuesto, aunque este método es muy simple, su uso se ve limitado por razones de eficiencia cuando el espacio de búsqueda es muy grande, aún para los esquemas mejorados de programación dinámica.

Los métodos aleatorios, por otro lado, han sido objeto de amplias investigaciones en la actualidad, ya que han tomado en cuenta los defectos de los esquemas basados en cálculo y los métodos enumerativos para no incluirlos. Dentro de estos métodos aleatorios se encuentran los Algoritmos Genéticos, los cuales usan opciones aleatorias como una herramienta para guiar el proceso de búsqueda a través de la codificación del espacio de búsqueda. Otra técnica es el recocido o templado simulado, el cual usa un proceso aleatorio para guiar la forma de búsqueda de estados mínimos de energía. Sin embargo, es importante señalar, que estas técnicas si bien es cierto usan métodos aleatorios, la búsqueda en realidad no es una búsqueda sin dirección.

b. Diferencias entre los Algoritmos Genéticos y los Métodos Tradicionales

Los Algoritmos Genéticos se diferencian de los métodos de búsqueda tradicionales debido principalmente a cuatro características:

- Los Algoritmos Genéticos trabajan con una codificación de un conjunto de parámetros, no con los parámetros mismos.
- Los Algoritmos Genéticos efectúan su búsqueda usando un conjunto de puntos y no un solo punto.
- Los Algoritmos Genéticos utilizan una función objetivo que guía el proceso de búsqueda. Por tanto, no son necesarias las derivadas de las funciones ni ningún otro parámetro que pueda ser tomado como una guía auxiliar en dicho proceso.

- Los Algoritmos Genéticos utilizan reglas de transición probabilísticas, no reglas determinísticas.

Los Algoritmos Genéticos necesitan un conjunto de parámetros naturales o valores del problema a optimizar, los cuales representan en sí una de las posibles alternativas de solución del mismo, sin que esto signifique que sean mínimas o máximas. Estos parámetros serán codificados en una cadena con un alfabeto finito, que las representará en las operaciones que se efectúen en el Algoritmo Genético. Cuando se inicia el algoritmo, se construyen un conjunto de las cadenas codificadas mencionadas anteriormente como posibles soluciones a la búsqueda y las cuales constituyen una población. Después de este inicio, se generan nuevas poblaciones siguiendo los mecanismos naturales de evolución y permitiendo localizar nuevas posibles soluciones al problema. De allí que el análisis siempre se refiere a un conjunto de puntos y no a uno solo, como en los demás métodos.

También las otras técnicas requieren información adicional para efectuar la búsqueda. Por ejemplo, los métodos de gradiente necesitan la derivada de la función objetivo para poder hacer las búsquedas de los picos locales. Por su parte los Algoritmos Genéticos, no necesitan ningún tipo de información auxiliar.

Todas estas diferencias permiten que los Algoritmos Genéticos sean métodos de búsqueda y optimización muy eficaces, cuando el espacio de estados posibles es muy grande o infinito.

3. Componentes de un Algoritmo Genético

Todos los Algoritmos Genéticos poseen componentes básicos para su funcionamiento como son una representación intrínseca al problema a resolver, la población inicial de soluciones que serán la base del algoritmo en un principio, la función de adaptación que permite evaluar que tan buena es una solución con respecto a otra, los operadores genéticos que permiten alterar la descendencia para localizar nuevas soluciones y los parámetros o valores que regirán la dinámica evolutiva de dicho algoritmo. Estos conceptos se verán con detalle a continuación.

a. Representación de las Soluciones del Problema

No todos los problemas a tratar por un Algoritmo Genéticos son iguales, por lo tanto, es necesario crear una estructura que se adapte a la generalidad de la ejecución de un Algoritmo Genético. Es por ello que se define una representación de las posibles soluciones de un problema como una cadena de símbolos de un alfabeto finito previamente seleccionado. Dicho alfabeto debe ser lo más pequeño y sencillo posible, pero que permita representar de manera exacta las características del problema a solucionar.

El alfabeto más utilizado para representar soluciones es el alfabeto binario, en donde toda solución puede ser definida como una cadena de ceros (0) y unos (1). Sin embargo, para diferentes tipos de problemas este alfabeto no es suficiente, por lo cual se hace uso de varios elementos en el mismo, solo con la regla general de que la combinación de los componentes del alfabeto produzca soluciones que sean válidas para el entorno del problema.

En la siguiente tabla se presentan varios conceptos de los sistemas naturales que a la vez son aplicados en los sistemas artificiales como los Algoritmos Genéticos y que permiten una equivalencia directa entre los mismos, además de ser usados indiferentemente:

Figura 6. Equivalencia de Términos en Sistemas Naturales y Artificiales

Sistemas Naturales	Sistemas Artificiales
Cromosoma	Cadena de símbolos del alfabeto seleccionado
Gen	Parámetro o característica del problema
Alelo	Valor de un parámetro
Locus	Posición de un gen en un cromosoma

b. Población Inicial de Soluciones

Previo a la ejecución de un Algoritmo Genético se debe crear un conjunto de posibles soluciones iniciales. Este conjunto de soluciones es también llamado población inicial y se convertirá en la base de la evolución del algoritmo. Esta población inicial, debe ser construida generando de manera aleatoria los valores de los genes para que constituyan un cromosoma y a su vez una solución válida al problema.

c. Función de Adaptación

La función de adaptación o también llamada función de calidad, es la que permite discriminar de manera efectiva un individuo sobre otro. Por lo tanto, se convierte en una medida de calidad de dicho individuo como posible solución para el problema propuesto.

Su importancia radica en que mediante esta función es posible categorizar los individuos y de esta manera, establecer a cuales de ellos se le aplicarán los métodos de selección, para posteriormente realizar el resto de operaciones que permitirán efectuar la evolución de los mismos, tal cual como los sistemas naturales lo realizan.

Sin embargo, es posible que con la función de adaptación definida se le esté dando mayor importancia a los mejores individuos de una población en detrimento de los demás. Por lo tanto, el material genético de estos últimos se perdería en los procesos de selección. Para evitar que esto ocurra, se utiliza la renormalización de los valores resultantes de la evaluación de la función de adaptación en cada individuo. La renormalización puede ser efectuada de dos maneras diferentes:

- Renormalización lineal: Esta renormalización consiste en ordenar los cromosomas en orden descendente dependiendo del valor obtenido al aplicarle la función de adaptación. Luego se le asigna un valor máximo al mejor de los individuos y posteriormente, se harán decrementos lineales a este valor para ser asignado al resto de los individuos dependiendo del orden que ocupan.
- Renormalización por Ventanas: En este tipo de renormalización se define una cota mínima para la función de adaptación. Posteriormente, a todos los individuos que estén por debajo de esa cota mínima, se les asigna ese valor y a los que estén por encima se les asigna la cantidad que constituya el exceso de esa cota.

d. Operadores Genéticos

Los operadores genéticos son los encargados de emular los procesos de reproducción natural que se presentan en los sistemas naturales.

Los operadores utilizados en la gran mayoría de los Algoritmos Genéticos son los de selección, cruce y mutación; aunque en ciertas oportunidades es posible encontrar operadores distintos a estos y que están asociados con la naturaleza del problema a resolver.

(1). Operador de Selección

El propósito de este operador es el de escoger los padres que aportarán su material genético para la creación de nuevos individuos. La condición más importante de este operador es la selección de padres que tengan los mejores valores de la función de adaptación.

Para ello se hace uso de un método de selección probabilístico, para asegurar que mientras mayor sea el valor de la función de adaptación del individuo, mayor será la probabilidad de escogerlo como un posible padre. Este método es conocido también como rueda de ruleta y consiste en realizar la selección de los padres utilizando un mecanismmo que simula a una rueda de una ruleta. En esta rueda de ruleta, cada individuo estará representado por un arco de la rueda total, y la longitud del arco de cada individuo en la ruleta será mayor o menor, dependiendo del valor de su función de adaptación. De allí que la probabilidad de que un individuo i sea escogido dependerá de la siguiente ecuación:

$$P_i = \frac{f_i}{\sum_{j=1}^{n} f_j}$$

donde f_i es el valor de la función de adaptación y

$$\sum_{j=1}^{n} f_j$$

es la suma total de todos los valores de las funciones de adaptación de todos los individuos que componen la población.

(2). Operador de Cruce

Este operador es el encargado de emular la reproducción sexual de los sistemas naturales. Mediante la aplicación del mismo es posible crear nuevos individuos a partir de los cromosomas de los padres. Los operadores de cruce más comunes son el cruce en un punto, el cruce en dos puntos y el cruce uniforme.

El operador de cruce en un punto consiste en seleccionar aleatoriamente una posición del cromosoma o punto de cruce e intercambiar el material genético a la izquierda de un padre con el de la izquierda del otro padre, para de esta manera crear dos individuos hijos que tendrán material genético de ambos padres. Este mecanismo se puede ejemplificar de la siguiente manera, sean X y Y los padres sobre los cuales se efectuará el cruce y los cuales se presentan a continuación:

$$X = X_1X_2X_3X_4X_5X_6X_7X_8X_9X_{10}$$
$$Y = Y_1Y_2Y_3Y_4Y_5Y_6Y_7Y_8Y_9Y_{10}$$

Si aleatoriamente se selecciona como punto de cruce el locus cuatro (4), los hijos resultantes tendrán la siguiente configuración:

$$X' = \mathbf{Y_1 Y_2 Y_3 Y_4} X_5 X_6 X_7 X_8 X_9 X_{10}$$
$$Y' = \mathbf{X_1 X_2 X_3 X_4} Y_5 Y_6 Y_7 Y_8 Y_9 Y_{10}$$

En el cruce en dos puntos por otro lado, se seleccionan dos posiciones aleatorias y luego se intercambia el material genético que este entre ambas posiciones, para dar como resultado hijos que poseen genes más variados pero siempre a partir de los padres. Así, si utilizamos los mismos padres mostrados anteriormente (X e Y) y seleccionamos los puntos de cruce cuatro (4) y seis (6), los nuevos individuos resultantes serán los siguientes:

$$X = X_1 X_2 X_3 \mathbf{Y_4 Y_5 Y_6} X_7 X_8 X_9 X_{10}$$
$$Y = Y_1 Y_2 Y_3 \mathbf{X_4 X_5 X_6} Y_7 Y_8 Y_9 Y_{10}$$

En el caso del cruce uniforme, se selecciona de manera aleatoria aquellos puntos los cuales serán intercambiados los genes de ambos padres. De esta manera si aleatoriamente se escogen los genes tres (3), cinco (5), seis (6) y diez (10) para intercambiar, los individuos resultantes serían:

$$X = X_1 X_2 \mathbf{Y_3} X_4 \mathbf{Y_5 Y_6} Y_7 Y_8 Y_9 \mathbf{X_{10}}$$
$$Y = Y_1 Y_2 \mathbf{X_3} Y_4 \mathbf{X_5 X_6} X_7 X_8 X_9 \mathbf{Y_{10}}$$

(3). Operador de Mutación

El operador de mutación crea nuevos individuos a partir de la alteración aleatoria de los genes de un individuo. En los sistemas naturales, la probabilidad de mutación de un gen es bastante pequeña y de esa manera debe utilizarse en los Algoritmos Genéticos. Sin embargo, en ciertos problemas si los individuos tienen probabilidad de mutación más alta, entonces existirá mayor diversidad en la población y por consiguiente, se podrá encontrar una mejor solución (convergencia del algoritmo).

Para implementar este operador se debe recorrer cada gen del cromosoma que representa al individuo y se escoge aleatoriamente si será objeto de mutación o no. Si se debe mutar dicho gen, entonces se deberá cambiar por otro valor que se encuentre dentro del alfabeto de la representación.

e. Parámetros de Operación de los Algoritmos Genéticos

Los parámetros de los Algoritmos Genéticos son valores que controlan el comportamiento del mismo y son lo más importante a la hora de analizar el rendimiento que este método puede alcanzar en la solución de problemas. Parte del éxito o no de la solución de un problema mediante el uso de Algoritmos Genéticos, involucra el poder manejar los mejores valores posibles para estos parámetros. Los parámetros más importantes son:

(1). Tamaño de la Población

El tamaño de la población afecta tanto al rendimiento global como a la eficiencia del Algoritmo Genético. Los Algoritmos Genéticos con poblaciones pequeñas actúan de manera pobre, ya que un tamaño reducido no provee una cobertura suficiente del espacio de soluciones del problema. En cambio, para poblaciones grandes hay una mejor representación del espacio de soluciones y se previene la convergencia prematura a soluciones locales. El valor más utilizado de este parámetro, según algunos estudios, es entre cuarenta (40) y cien (100) individuos.

(2). Probabilidad de Cruce

La probabilidad de cruce controla la frecuencia con la cual se aplica el operador de cruce. Para este parámctro suele utilizarse un valor del 85% o un valor mayor. En cada nueva generación, sólo este porcentaje de individuos sufre un cruce. Si la probabilidad de cruce es muy alta, los individuos más aptos pueden destruirse con más rapidez con la que se crean, haciendo lenta la estabilización o convergencia del algoritmo. Si la probabilidad de cruce es baja, la población podría estancarse e irse acercando a una homogeneidad de individuos posiblemente alejados del óptimo.

Este parámetro guarda una estrecha relación con el tamaño de la población. Se recomiendan altas probabilidades de cruce para poblaciones pequeñas, un amplio rango de probabilidades se sugieren para poblaciones intermedias y valores bajos de este parámetro sólo para poblaciones grandes.

(3). Probabilidad de Mutación

La probabilidad de mutación controla la frecuencia con la cual se aplica el operador de mutación. Cada gen del individuo tiene una probabilidad finita de mutar. Una probabilidad de mutación baja previene el congelamiento de los individuos en un valor particular y esta debe ser baja en las primeras generaciones para permitir que la población se estabilice, logrando un equilibrio dinámico. En las etapas finales del proceso evolutivo, se puede utilizar un incremento en la probabilidad a fin de acelerar la variación genética de la población y tener la certeza de haber realizado exploraciones en el espacio de soluciones en su totalidad.

(4). Elitismo

Para producir una convergencia más rápida y mejorar el desempeño del Algoritmo Genético, se utiliza también el criterio de escoger un conjunto de los mejores individuos y trasladarlos directamente a la próxima generación. Este mecanismo es llamado elitismo, y consiste en definir un valor porcentual de individuos, los cuales pasarán a la próxima generación sin que se ejecuten sobre ellos ningún tipo de operador genético. Con el elitismo se consigue una mejora en la ejecución del algoritmo, ya que se disminuye la cantidad de operaciones computacionales y por consiguiente, se mejora también la convergencia, ayudando a su vez a evitar que se cometan errores estocásticos como en el caso de que en generaciones intermedias se consigan mejores individuos que en las generaciones finales.

4. Algoritmos Genéticos y Autómatas Celulares: Autómatas Celulares Evolutivos

Los Autómatas Celulares Evolutivos son un marco ideal para el estudio de la manera como la evolución, bien sea natural o artificial, puede crear sistemas en los cuales la computación emergente toma lugar, según Mitchell et. al. (1996). En este caso, los componentes simples con información local y comunicación entre ellos interactúan para coordinar el procesamiento general de la información. Las colonias de insectos, sistemas económicos, el sistema inmunológico y el cerebro son sistemas donde esta computación emergente siempre ocurre. No obstante, no se conoce a ciencia cierta como se hacen estos cálculos en los mencionados sistemas.

El mecanismo como se implementan los Autómatas Celulares Evolutivos implica tomar como base un Algoritmo Genético para efectuar la búsqueda de una tabla de reglas de Autómatas Celulares de cierta vecindad. Para ello, se generan en la población, cromosomas que representan reglas candidatas que consisten en los bits de salida de la tabla de dicha regla. De esta manera en el caso de las reglas para vecindad de una celda (r=1), los cromosomas serán de 8 bits, para r=2 serían de 32 bits, para r=3 serían de 128 bits y así sucesivamente. De este modo, el tamaño del espacio de búsqueda será hasta de 2^{128} reglas en el último de los casos, lo cual es bastante amplio como para poder utilizar algún método de búsqueda exhaustiva y es por ello que se hace uso de otros métodos como los Algoritmos Genéticos.

C. Compresión de Imágenes

1. Introducción a los Conceptos de Compresión

La compresión de datos en general puede ser definida como el arte o la ciencia capaz de representar la información de manera compacta, Sayood (1996). Para la creación de esa representación compacta, se deben localizar y hacer uso de las estructuras existentes en los datos mismos. Los datos pueden ser caracteres de un archivo de texto, números que son ejemplos de ondas de voz o imágenes y hasta secuencias de números que son generados por una gran variedad de procesos.

Un ejemplo de la compresión de datos viene representado por el código Morse, que fue utilizado a mediados del siglo 19. Las cartas que eran enviadas por el telégrafo eran compuestas de una serie de líneas y puntos. Samuel Morse notó que había un conjunto de letras del alfabeto que se repetía con mayor frecuencia que otras, por lo tanto asignó secuencias más cortas de puntos y líneas a los mismos. De esta idea comenzaron a surgir codificaciones para la compresión de datos basadas en la frecuencia de aparición de caracteres.

Otro ejemplo de compresión es el código Braile de Grado 1, también desarrollado a mediados del siglo 19, el cual usa la frecuencia de palabras en las frases para efectuar la compresión. Este método utiliza una rejilla de dos (2) por tres (3) puntos los cuales pueden estar en relieve o no. Por lo tanto con seis (6) puntos diferentes, cada uno de los cuales poseen dos (2) estados, se pueden obtener hasta 64 combinaciones diferentes. En el caso del código Braile de Grado 2, se utilizan las combinaciones sobrantes que no representan

letras del alfabeto para representar palabras completas, lo cual reduce la cantidad de rejillas necesarias al escribir hasta en un 20% en promedio, Bell et al (1990).

Lo que en los dos ejemplos anteriores se especificó, no obstante, son estructuras estáticas para realizar la compresión, pero ese no es el único tipo de estructura que existe en los datos necesariamente. Hay muchos tipos de estructuras en los datos que pueden ser comprimidas también, como por ejemplo los datos de voz, los cuales pueden traducirse en números que representan la configuración sonora.

Aún cuando existen otras estructuras mediante las cuales se pueden representar los datos para realizar su compresión, una estructura no es la única característica que poseen los datos que permite efectuar dicho proceso. También se puede utilizar el hecho de que los datos serán utilizados por los sentidos humanos, los cuales no son en muchas ocasiones capaces de percibir en totalidad los rasgos más detallados de los datos como las imágenes y la voz. Esto es explotado por algunos mecanismos de compresión como es el caso del presentado en este trabajo.

2. Técnicas de Compresión

Para el desarrollo de un algoritmo de compresión es necesario cumplir con dos etapas diferenciadas: la creación del modelo y la codificación.

a. Creación del Modelo

En la primera fase o creación del modelo, se intenta extraer cualquier información redundante en los datos, posteriormente la descripción de esa redundancia se convertirá entonces en el modelo. Los modelos más ampliamente utilizados en la compresión son el Modelo Estadístico y el Modelo basado en Diccionarios.

(1). Modelo Estadístico

El Modelo Estadístico en su forma más sencilla, utiliza una tabla estática de probabilidades en donde a cada símbolo se le calcula la probabilidad de ocurrencia dentro de los datos a ser comprimidos. Esta tabla estadística esta representada por una estructura tipo árbol conocida como árbol de Huffman. Una vez realizado esto, se almacena la tabla estadística y una versión codificada de los datos, lo cual produce la compresión. Esta tabla estadística es única para cada conjunto de datos diferentes que hayan sido comprimidos, por lo que al añadir nuevos datos a los previamente comprimidos es necesario crear de nuevo la tabla estadística desde un principio. Para el caso de este modelo, se utiliza muy frecuentemente también el modelo de cadenas de tiempo discretas de Markov.

(2). Modelos basados en Diccionarios

Los Modelos Estadísticos generalmente codifican un solo símbolo a la vez, calculando la probabilidad del mismo en los datos para luego codificarlo. En cambio, en los modelos basados en diccionarios se leen los datos de entrada y se localizan grupos de símbolos que aparecen en el diccionario. Si una cadena de caracteres es localizada,

entonces la salida codificada sería un apuntador a la entrada del diccionario que representa dicha cadena, en vez de la cadena misma.

Tal es el caso del método conocido como LZW y que fue presentado por primera vez por Lempel y Ziv (1977) y que es conocido también como LZ77. Este método fue mejorado un año después por los mismos autores, Lempel y Ziv (1978) y fue llamado LZ78.

En el caso del LZ77, el diccionario consiste de todas las cadenas presentes en un espacio predefinido o ventana perteneciente a los datos de entrada. Por ejemplo, un programa de compresión basado en este esquema podría utilizar una ventana de dos (2) Kilobytes como diccionario. Así, mientras se leen nuevos grupos de símbolos de los datos de entrada, el algoritmo busca en la ventana previamente definida los patrones coincidentes. Cualquier coincidencia se codifica entonces como un apuntador.

Este modelo produce algoritmos de compresión bastante eficientes y han sido utilizados como base para nuevos algoritmos en programas comerciales como PKZIP y LHarc.

Por otro lado, el algoritmo LZ78 toma un diferente alcance en lo que se refiere al mantenimiento y creación del diccionario mismo. En vez de tener una ventana de tamaño fijo y limitada, el algoritmo crea su diccionario de todos los símbolos previamente vistos en los datos de entrada. Este es un proceso incremental, en donde el diccionario se

construye tomando un carácter a la vez de una cadena de caracteres simple, y a medida que el algoritmo avanza, se añaden más caracteres a dicha cadena, hasta lograr conseguir una cadena que se repita y es entonces cuando dicha cadena es añadida al diccionario.

Este procedimiento trabaja muy bien cuando hay una gran cantidad de cadenas que se utilizan frecuentemente dentro de los datos de entrada. No obstante, a diferencia del método LZ77, las cadenas de caracteres en el diccionario del método LZ78 pueden ser extremadamente largas, lo cual disminuye su capacidad de compresión. Este algoritmo es el utilizado para comprimir en los Sistemas Operativos UNIX.

En el caso de la compresión de imágenes, los modelos descritos anteriormente no son muy efectivos, ya que cuando las imágenes poseen tonos muy similares no logran efectuar una buena compresión y además los diccionarios o tablas estadísticas deben acompañar a los datos codificados, por lo que aumenta el tamaño de los datos luego de la compresión.

b. Codificación

La segunda fase o codificación, consiste en la descripción del modelo y de cómo se diferencian los datos del modelo planteado. Esto último se hace regularmente usando un alfabeto binario. La diferencia existente entre los datos y el modelo es conocida también como el residuo.

3. Esquemas de Compresión

Cuando se habla de una técnica de compresión, en realidad se están hablando de dos algoritmos diferentes. Está el algoritmo encargado de tomar los datos de entrada X y generar una representación X_c que requiere menos bits para representar la información, y está el algoritmo de reconstrucción que toma la representación comprimida de los datos X_c y los transforma en datos de salida Y.

Basado en las necesidades de reconstrucción de los datos, los esquemas de compresión se dividen en dos clases: esquemas de compresión sin pérdidas o "Lossless", en donde Y es exactamente igual a X y esquemas de compresión con pérdidas o "Lossy", en donde por lo regular se puede obtener mejor compresión pero Y es ligeramente diferente a X.

a. Compresión sin Pérdidas o "Lossless"

En este esquema los datos previamente comprimidos se recuperan de una manera exacta, por tanto es ampliamente utilizado para datos discretos como texto, datos generados por computadoras y algunos tipos de información de imágenes y videos.

Si los datos deben ser procesados y mejorados posteriormente para utilizar más información, es importante que la integridad de la misma se mantenga. Por ejemplo, cuando se comprimen imágenes de satélite para luego reconstruirlas y procesarlas a un tamaño mayor, es evidente que si la imagen no es fiel a la original, esas diferencias pueden incidir en una representación distorsionada del nivel de detalle al cual se quería llegar.

b. Compresión con Pérdidas o "Lossy"

Este esquema de compresión implica alguna pérdida de información, ya que los datos que han sido comprimidos de esta manera no pueden ser reconstruidos exactamente. A cambio de esta distorsión en la reconstrucción, se pueden obtener porcentajes de compresión mucho más altos que el obtenido por el esquema de compresión previamente definido.

En muchas aplicaciones esta falta de información no es relevante. Por ejemplo, cuando se transmite la voz, no es necesario transmitir el valor exacto de la misma. Dependiendo de la calidad necesaria se reconstruye la voz, pudiendo variar los porcentajes de pérdidas y aún se podría entender totalmente lo que allí esta almacenado. Ese rango podría variar, por ejemplo, entre la calidad de la voz en una transmisión telefónica y la calidad en un disco compacto, donde solo se tolera una perdida de información muy pequeña.

Hasta hace poco este tipo de compresión solo se utilizaba en dispositivos electrónicos o hardware especializado. Pero con la disminución de precios de los Procesadores Digitales de Señales (DSP), se ha venido expandiendo su uso en diversas aplicaciones computacionales como es el caso de los circuitos para el tratamiento del sonido.

Otra característica de este tipo de compresión, es que frecuentemente se realiza en dos pasos. Primero se aplica a un alto nivel una función de procesamiento de señales

sobre los datos, lo cual consiste en la transformación de los datos a un dominio de frecuencia usando algoritmos similares a la transformada de Fourier (FFT). Una vez que los datos han sido transformados, estos se suavizan eliminando los puntos más altos y más bajos de la transformación y es en este paso donde se produce la pérdida de información. Por último, se aplica una compresión normal sin pérdidas a los datos transformados, lográndose el producto final de la compresión.

Basado en este mecanismo y con la anuencia del Comité Internacional de Estándares (ISO) y el Comité Consultivo Internacional para la Telegrafía y Telefonía (CCITT), se han desarrollado dos comités para la fijación de estándares de compresión de imágenes y videos como son el "Joint Photographic Expert Group" (JPEG) y el "Moving Picture Expert Group" (MPEG).

(1). El Modelo "Joint Photographic Expert Group" o JPEG

El estándar JPEG usa el algoritmo de la Transformación Discreta del Coseno (DTC) para convertir un gráfico a un dominio de frecuencia. El JPEG establece también varios niveles o factores de calidad entre 0 y 100 y permite al mecanismo de compresión el elegir que factor utilizará. Por lo general, los promedios de compresión de este algoritmo son entre el 90 y el 95 %, con poca o ninguna degradación.

La especificación JPEG consta de muchas partes, incluyendo las referentes a la codificación "Lossy" y "Lossless". La compresión "Lossless" utiliza un modelo

predictivo/adaptativo con codificación Huffman, el cual produce una buena compresión de imágenes sin la pérdida de su resolución.

En cuanto a la compresión "Lossy", el estándar utiliza la Transformación Discreta del Coseno (DCT), la cual está dentro de las clases especiales de funciones matemáticas como la Transformada de Fourier (FFT), que permiten tomar una señal y transformarla de un tipo de representación a otro.

(2). Compresión Fractálica

La compresión JPEG es muy efectiva en relaciones de compresión de entre 20:1 y 25:1. Pero más allá de este punto, la imagen comienza a formar bloques de colores, lo cual provoca la pérdida de la calidad de la imagen inicial impidiendo su uso práctico. Además, a altas relaciones de compresión el método JPEG introduce artefactos en la imagen, en especial en los bordes difusos de la misma, lo cual le resta también calidad. De allí que han aparecido algunos otros métodos como la Cuantificación de Vectores, los métodos basados en ondas o la Compresión Fractálica.

El término fractal fue utilizado por primera vez por Benoit Mandelbrot para designar objetos que son similares pero a diferentes escalas. Un ejemplo muy conocido es el fractal que posee el nombre de conjunto Mandelbrot, el cual esta representado por una sola ecuación. Sin embargo, los fractales no fueron considerados para la compresión de datos hasta el desarrollo de la teoría de Sistemas de Funciones Iteradas (IFS) en 1981 por J. Hutchinson. Esta teoría consiste en localizar un conjunto de funciones, las cuales al ser

aplicadas una tras otra a los datos comprimidos, permiten obtener la imagen inicial previamente definida. Con esta teoría se lograban compresiones de hasta 10000:1, pero lamentablemente solo podían obtenerse esa compresión en imágenes previamente elaboradas y no en imágenes generales, además de requerir la ayuda de una persona en el proceso de compresión mismo.

De allí surge la idea en 1988 de Arnaud Jacquin que consistía en sustituir la búsqueda de una IFS que representara la imagen, por la división de dicha imagen en rangos no sobrepuestos y encontrar el IFS para cada uno de los rangos. Este proceso también es conocido como el Sistema de Funciones Iteradas Particionadas (PIFS) y es la base de la compresión fractálica en sí. Sin embargo, este tipo de compresión aún sigue siendo objeto de una activa investigación.

4. Eficiencia del Método de Compresión

Una vez que se desarrolla un esquema de compresión de datos, es necesario medir su eficiencia. Debido al gran número de áreas de aplicación, se han desarrollado diferentes términos para describir y medir esta característica.

Un algoritmo de compresión puede ser evaluado de muchas maneras, bien sea según la complejidad del algoritmo, la memoria necesaria para implementar el algoritmo, cuan rápido se efectúa la compresión en un equipo determinado, el tamaño final de la compresión y que tan cerca esta la reconstrucción de los datos del original. De estas medidas las más importantes son exactamente el tamaño final de la compresión y la

similitud de los datos construidos del original, ya que generalizan y diferencian cada método de los demás.

Una excelente manera de medir que tan bueno es un algoritmo al comprimir un conjunto de datos, es buscar la relación del número de bits requeridos para representar los datos antes de la compresión con respecto al número de bits requeridos para representar los datos luego de la compresión. Esta relación es llamada relación de compresión o rata de compresión. Suponiendo que se tiene una imagen de 256 por 256 pixeles, se necesitarán 65536 bytes para almacenarla. Esta imagen es comprimida y necesitando 16384 bytes para almacenarse, entonces se puede decir que la rata o relación de compresión es de 4:1.

Otra manera de reportar la eficiencia de la compresión es dar el número promedio de bits necesario para representar un conjunto de datos. Por ejemplo, en el caso de la imagen comprimida definida anteriormente, si se asumen ocho bits por pixel, entonces el número promedio de bits por pixel en la representación comprimida es de dos (2). De esta manera, se puede decir que el promedio es de dos (2) bits por pixel.

En la compresión con pérdidas o "Lossy", debido a que la reconstrucción de los datos difieren del original, para medir la eficiencia de un algoritmo de compresión se puede cuantificar esta diferencia que también es llamada distorsión. Otros términos empleados para definir esta característica son también fidelidad, aproximación y calidad. Cuando se dice que la fidelidad, aproximación o calidad de una imagen es alta, entonces indica que la diferencia entre la reconstrucción y la imagen original es muy pequeña. Esta diferencia

puede ser tanto matemática como perceptiva, pero en todo caso son las medidas matemáticas las utilizadas para hacer la comparación entre los algoritmos.

III. DESARROLLO Y EJECUCION DE LA SOLUCION PROPUESTA

A. Análisis y Características de la Solución Propuesta

La solución general del problema consiste en obtener un conjunto de Autómatas Celulares y operadores binarios, los cuales al aplicarse sucesivamente, terminan por reconstruir una imagen que en el peor de los casos posea una gran semejanza con la imagen original.

Para ello, se deben definir algunas características de la evolución de los Autómatas Celulares como son los estados iniciales, el transiente a tomar en cuenta en la evolución de los mismos y otras. También es importante definir las características internas de operación del Algoritmo Genético, como lo es la codificación binaria de la solución en un cromosoma, que función de adaptación se debe considerar para la evaluación de los individuos, el tipo de operador de selección, cruce y mutación y cualquier otra característica relevante para el diseño del Algoritmo Genético.

1. Codificación Binaria

Para representar los Autómatas Celulares y sus operadores se hace uso de un cromosoma binario de 34 bits, que permiten encapsular de una manera única los dos datos necesarios para definir a cada individuo: el operador booleano y la regla de evolución del Autómata Celular. Los datos a ser representados en el cromosoma binario se pueden definir de la siguiente manera:

$$x = (\alpha 1, \alpha 2, \beta 1, \beta 2, \ldots, \beta 32) \text{ donde } \alpha, \beta \in \{1,0\}$$

En este caso se hace uso de los dos (2) primeros bits ($\alpha 1$, $\alpha 2$) para representar los operadores binarios, según la tabla de decodificación que se muestra a continuación:

Figura 7. Equivalencia de operadores binarios

Operador binario	$\alpha 1$	$\alpha 2$
AND	0	0
OR	0	1
XOR	1	0

En el caso de los restantes 32 bits, estos representa la regla de evolución del Autómata Celular, con una vecindad de r=2.

Es importante resaltar, que aquellos cromosomas cuyos dos primeros bits sean unos, serán tomados como cromosomas no válidos para la ejecución del algoritmo y por lo tanto, no serán incluidos en la población a evaluar como un nuevo componente de la misma. También existe una condición adicional en la primera ejecución del algoritmo, ya que en ese caso por ser necesario buscar el Autómata Celular inicial que será el primer término de las sucesivas operaciones booleanas, solo es necesario obtener el número que representa la regla de evolución del Autómata Celular y no es relevante el operador booleano representado por los dos primeros bits de la representación.

2. Autómatas Celulares

El objetivo principal del Autómata Celular es crear patrones que mediante operaciones booleanas, producirán una aproximación a una imagen objetivo de 256 por 256 pixeles. Para ello, como estado inicial de todos los Autómatas Celulares, se tomó una cadena de 256 bits con valores de unos y ceros alternos (101010....10). Es importante señalar que la condición de borde aplicada a este Autómata Celular permite definirlo como un autómata circular, ya que el primer y último bit se considerarán células vecinas.

Sobre este estado inicial, el Autómata Celular debe evolucionar, aplicándole la regla dinámica codificada que está representada por los individuos de la población definida en el Algoritmo Genético y la cual se iterará por 276 etapas de tiempo. Ya que se considera que la dinámica del transiente de la evolución del Autómata Celular está incluida en las etapas de tiempo iniciales, se eliminan las primeras 20 etapas de tiempo y se toman solo aquellas etapas desde la 21 hasta la 276.

Este patrón generado por el Autómata Celular, luego es comparado contra la imagen objetivo mediante la función de adaptación que será explicada posteriormente. Por tanto, al finalizar la ejecución del primer Algoritmo Genético (luego de 20 generaciones), se obtendrá como mejor individuo una regla dinámica que al evolucionar el Autómata Celular, generará un patrón que posee la mayor similitud con la imagen objetivo.

Posteriormente a la primera ejecución del Algoritmo Genético, se realizan ejecuciones sucesivas del mismo. Por cada nueva ejecución se obtendrá una operación

booleana y una regla dinámica. Esta regla dinámica debe evolucionar al Autómata Celular y generar un nuevo patrón, que al componerse con los patrones de imágenes previamente encontrados mediante la operación booleana, generará una nueva aproximación a la imagen objetivo.

Al finalizar el algoritmo de compresión, se obtendrán una serie de reglas dinámicas y operadores booleanos que las asocian, para generar una aproximación a la imagen objetivo. Por último, el criterio de parada del algoritmo de compresión se dará cuando se obtenga una similitud de 99% o más con respecto a la imagen objetivo, o luego de la ejecución de por lo menos 20 Algoritmos Genéticos.

3. Función de Adaptación

Para el desarrollo de la función de adaptación se tomaron en cuenta varias alternativas que permitieran medir la similitud entre la imagen objetivo y la composición booleana de patrones que se ha alcanzado al finalizar el algoritmo. Para ello se utilizó una función que mide la cantidad de bits que coinciden en el patrón generado y en la imagen objetivo y se relacionan con la cantidad de bits que no coinciden entre ambas imágenes. La fórmula, es presentada a continuación:

$$\frac{(BB + NN)}{(BN+NB+1)}$$

donde BB es el total de pixeles cuyo color blanco coincide en la imagen objetivo y en la aproximación generada, NN es el total de pixeles negros coincidentes, mientras que BN y NB representan los pixeles que no son coincidentes.

Se hace uso de esta función ya que con ella se castiga a los individuos que posean un menor número de aciertos en los pixeles de la imagen compuesta con relación a la imagen objetivo. De esta manera, cuando existe un mayor número de aciertos, la función tiende a la distancia de Hamming, mientras que al existir un menor número de aciertos, la función tiende a ser cero. De esta manera, los individuos con mayor similitud tendrán un mayor valor de adaptación.

4. Operador de Selección

Para el operador de selección se hizo uso del mecanismo de rueda de ruleta ("roulette wheel") explicado anteriormente. Sin embargo, la función de adaptación tiende a favorecer mucho a los individuos con mayor valor de adaptación en detrimento de aquellos que poseen menor valor de adaptación, como pasa frecuentemente en este tipo de problemas. Por lo tanto, y para evitar que se creen nuevos individuos sin hacer uso de aquellos que menor valor de adaptación posean, lo cual provocaría una menor diversidad en la nueva población, se hizo uso de la renormalización lineal de la adaptación real. Los valores utilizados para la renormalización lineal fueron un valor máximo de 100 y un decremento de 1 para cada uno de ellos.

5. Operador de Cruce

Para el operador de cruce se seleccionó el cruce en un solo punto, en donde la posición en la cual debe efectuarse el cruce, entre la posición 1 y la 34, dependerá de un valor aleatorio.

B. Organización de los Programas

Para el desarrollo de los programas se escogió el lenguaje de programación JAVA, tomando en consideración las características de este lenguaje, en cuanto a las facilidades que provee para ser portado a diferentes plataformas, la posibilidad de la definición de objetos para el tratamiento de la información y el hecho de proveer un entorno totalmente multitarea con niveles de paralelismo. Esta última facilidad permite que se pueda establecer a nivel del programa, una de las características de los Autómatas Celulares como es el ser sistemas intrínsecamente paralelos.

Se desarrollaron dos (2) programas, los cuales están relacionados con el proceso de compresión de la imagen y la descompresión de la misma. En el caso de la compresión de la imagen, el programa se compone de la ejecución de varios Algoritmos Genéticos que tienen como función el tratar de conseguir una sucesión de reglas de evolución de Autómatas Celulares que generen imágenes parciales y operadores booleanos, que al ser ejecutados sobre estas imágenes parciales generarán la imagen previamente comprimida.

En el caso del segundo programa, su función será la de implementar el mecanismo de descompresión, con lo cual tomará el resultado del programa de compresión (reglas y

IV. EXPERIMENTACION

La etapa de experimentación de este trabajo se divide en dos partes. En la primera parte, se deben conseguir los parámetros del Algoritmo Genético que sean más adecuados para la obtención de la mejor aproximación porcentual entre la aproximación resultante y la imagen objetivo que sufrió el proceso de compresión. En la segunda parte, se tomará el Algoritmo Genético con los parámetros previamente encontrados y se realizará una comparación con los métodos de compresión de imágenes más ampliamente utilizados

A. Selección de los Parámetros del Algoritmo Genético

Los parámetros que se seleccionaron para el Algoritmo Genético fueron la cantidad de la población inicial, la probabilidad de cruce, la probabilidad de mutación y la cantidad de generaciones. Para seleccionar los mejores parámetros que serán utilizados en la ejecución final de la experimentación, se tomaron en consideración los valores que se resumen en la siguiente figura y que pueden ser considerados como valores ampliamente utilizados en este tipo de problemas:

Figura 8. Parámetros a ser evaluados en la fase de experimentación

Parámetro	Valores evaluados
Población inicial	40 y 100
Probabilidad de cruce	0.80, 0.85 y 0.90
Probabilidad de mutación	0.01, 0.03, 0.05 y 0.10
Cantidad de generaciones	10 y 20