

Efecto del tamaño del archivo, la entropía y el tamaño del alfabeto en el rendimiento del algoritmo de Huffman

Carlos Rincón, Alfredo Acurero, David Bracho y Juan Jakymec*

Unidad de Redes e Ingeniería Telemática, Departamento de Computación, Facultad Experimental de Ciencias, Universidad del Zulia, Venezuela.

Recibido: 28-01-08 Aceptado: 03-09-08

Resumen

El presente trabajo tuvo como finalidad determinar el rendimiento del algoritmo de Huffman bajo la variación de los parámetros independientes: tamaño del alfabeto, tamaño del archivo y entropía. Las variables dependientes utilizadas para determinar el rendimiento del algoritmo fueron la relación de compresión, el tiempo de compresión y el tiempo de descompresión. El diseño seleccionado para realizar el análisis estadístico fue un modelo de bloques al azar, utilizando el tamaño del alfabeto como factor de bloqueo y los tratamientos (tamaño del archivo y entropía) en un arreglo factorial 3x2, y aplicando el procedimiento de análisis de varianza. El análisis de los resultados determinó que para la relación de compresión todas las variables independientes tienen un efecto significativo, mientras que para el tiempo de compresión y el tiempo de descompresión solo tuvieron un efecto significativo el tamaño del alfabeto y el tamaño del archivo.

Palabras claves: rendimiento, compresión, Huffman.

Efect of file size, entropy and alphabet size on Huffman algorithm performance

Abstract

The purpose of the present work was to determine Huffman algorithm performance under the variation of the independent parameters: alphabet size, file size and entropy. The dependent variables used to determine the algorithm performance were compression ratio, compression time and decompression time. The selected experimental design to do the statistical analysis was a randomized block model using the alphabet size as a blocking factor and the treatments (file size and entropy) in a 3X2 factorial array, applying the anova procedure. The results of the statistical analysis showed that for the compression ratio, all the independent variables have a significative effect, while for the compression time and the decompression time only the alphabet size and the file size had a significative effect.

Key words: performance, compression, Huffman.

* Autor para la correspondencia. E-mail: crincon@luz.edu.ve.

Introducción

Desde que los seres humanos tienen la capacidad de analizar lo que ocurre en su entorno, se ha generado información. Muchas ciencias definen la información; sin embargo, no fue sino hasta mediados del siglo pasado, como consecuencia de la aparición de la computación, cuando se establecieron los mecanismos para cuantificar la información.

A pesar de que los sistemas de cómputo tienen como tarea fundamental generar información, la mayoría de los estudiantes y egresados en el área de la computación definen la información como un conjunto de datos que, estructurados, tienen un significado, y solo saben que los computadores almacenan o transmiten la información utilizando la unidad mínima llamada bit.

En 1942, Claude Shannon (1) se encargó de definir un método matemático que permitió cuantificar la información generada por una fuente de datos. Shannon logró determinar lo que hoy llamamos teoría de la información, utilizando un concepto relativamente sencillo: considerando que todo evento tiene una probabilidad de ocurrencia, la cantidad de información producto de este evento puede cuantificarse mediante una función representada como la inversa de la probabilidad de ocurrencia de dicho evento. De este concepto se concluye que un evento con menor probabilidad de ocurrencia generará mayor información que un evento con mayor probabilidad de ocurrencia.

La necesidad del ser humano de procesar mayor cantidad de datos ha causado un problema debido a que se ha maximizado la cantidad de información generada. La información debe ser almacenada y transmitida, por lo que desde el punto de vista computacional se ha incrementado en el tiempo la capacidad de almacenamiento y de transmisión de los sistemas informáticos. Sin embargo, la cantidad de información generada excede la capacidad de los sistemas de cómputo, lo que se traduce en la necesidad de establecer me-

canismos que permitan aprovechar de manera efectiva los recursos computacionales.

Para resolver el problema planteado, la ciencia de la computación utilizó las mismas técnicas aplicadas en el pasado para resolver problemas similares. Mediante la codificación eficiente de los símbolos se obtiene la representación de la información original utilizando una menor cantidad de unidades de información.

La compresión de datos es el proceso que permite encontrar la codificación óptima para representar los datos generados por una fuente, con la finalidad de minimizar la cantidad de información producida por esta. Según Sayood (2), los algoritmos de compresión se clasifican en algoritmos con pérdida (cuando la información que resulta del proceso de compresión no es exactamente igual a la información original) y algoritmos sin pérdida (cuando la información que resulta del proceso de compresión es exactamente igual a la original). El uso de técnicas de compresión con o sin pérdida dependerá de las características de la información a comprimir.

A principios de los años cincuenta, David Huffman (3), alumno de Robert Fano (compañero de Shannon), en respuesta a una asignación en el MIT, planteó una técnica que permite generar códigos óptimos para realizar el proceso de compresión. Esta técnica de compresión sin pérdida se conoce en la actualidad como el algoritmo de compresión de Huffman, el cual mantiene su vigencia a pesar de la introducción de nuevos algoritmos para generar códigos óptimos.

Existen diferentes parámetros que afectan el rendimiento del algoritmo de Huffman. Algunos de estos parámetros son: (a) el tamaño del archivo, que se define como la cantidad de información que se encuentra en el archivo original; (b) la entropía, que se define como el tamaño promedio esperado de los códigos generados; y (c) el tamaño del alfabeto, que se define como la cantidad de símbolos que pueden ser utilizados para generar información.

El propósito del presente trabajo consistió en determinar el efecto que causan los parámetros independientes, tamaño del archivo, entropía y tamaño del alfabeto, en el rendimiento del algoritmo de compresión de Huffman.

Motivación

El algoritmo de compresión de Huffman fue diseñado en 1952 por David Huffman, quien lo fundamentó en la teoría de la información propuesta en 1942 por su profesor en el Instituto Tecnológico de Massachussets, Claude Shannon. A pesar del tiempo que ha pasado desde su diseño y el desarrollo de nuevos algoritmos de compresión de datos, Abraham Bookstein (4), en su investigación titulada *Is Huffman coding dead?*, concluyó que el algoritmo de Huffman puede utilizarse actualmente para resolver diferentes problemas informáticos que requieran de la compresión de datos.

Una de las aplicaciones en las que actualmente se utiliza el algoritmo de compresión de Huffman es la académica, como lo demuestran los trabajos de Mohamed Hamada (5) y James Keeler (6), en los que se implementa el algoritmo de Huffman para demostrar los conceptos implícitos en la teoría de la información.

Algunos de los factores que condicionan la efectividad del algoritmo de Huffman son el tamaño del archivo, la entropía y el tamaño del alfabeto. Basados en la teoría de la información, Edleno Silva de Moura et ál. (7) definen la entropía como la sumatoria de $i = 1$ hasta n (donde n es el tamaño del alfabeto) de la probabilidad de ocurrencia del símbolo i multiplicado por la información generada por el símbolo i , donde la información se define como el logaritmo base 2 del inverso de la probabilidad de ocurrencia del símbolo. Desde el punto de vista conceptual, la entropía se define como el tamaño promedio esperado de los códigos generados por la aplicación del algoritmo de Huffman.

El propósito de la investigación planteada consistió en medir el efecto de la variación del tamaño del archivo, la entropía y el tamaño del alfabeto en el rendimiento del algoritmo de Huffman, con la finalidad de poder identificar, con un basamento teórico, las aplicaciones en las que el algoritmo estudiado responde de manera eficiente.

Algunas aplicaciones del algoritmo de Huffman

El algoritmo de Huffman mantiene su vigencia como consecuencia de su aplicación en distintas tareas que requieren de la compresión de datos, así como fundamento de nuevas soluciones de compresión. Algunas de estas aplicaciones son:

Modified JPEG Huffman coding (8): en esta investigación Gopal Lakhani plantea una modificación a la codificación de Huffman utilizada por el algoritmo de compresión base de JPEG. Los resultados experimentales mostraron una reducción de 4% en la relación de compresión, comparados con el método de codificación de Huffman en modo secuencial y el método de codificación aritmético presentes en JPEG.

An efficient method of Huffman decoding for MPEG-2 AAC and its performance analysis (9): en esta investigación, Lee et ál. presentan un nuevo método de decodificación de Huffman especialmente diseñado para el formato de audio MPEG-2 ACC. El método mejora significativamente la eficiencia de procesamiento del algoritmo de decodificación de Huffman convencional que utiliza el mecanismo de búsqueda en árboles binarios. Los resultados experimentales demostraron un incremento promedio de 67% y 285%, comparado con el método de búsqueda en árboles binarios y búsqueda secuencial, respectivamente. También el método propuesto mostró una mejora en la eficiencia de procesamiento, dado que requirió menor cantidad de espacio en la memoria en relación con los nuevos mecanismos de búsqueda de Hashemian y sus variantes.

Direct mapping architecture for JPEG Huffman decoder (10): en esta investigación, Chang et ál. presentan un nuevo algoritmo para la decodificación de códigos de Huffman en el formato JPEG, en el cual la tabla de Huffman es dividida en cuatro módulos. Considerando esta partición una técnica de mapeo directo, fue desarrollada para decodificar los códigos de Huffman sin utilizar operaciones de búsqueda o comparación. Como resultado se obtiene que el algoritmo propuesto tiene un mejor desempeño que el algoritmo convencional.

Security analysis of multimedia encryption schemes based on multiple Huffman table (11): en esta investigación, Zhou et ál. plantean los problemas de seguridad presentes en los esquemas de encriptación de información multimedia basados en las tablas múltiples de Huffman. Este trabajo muestra otra aplicación diferente a la compresión de datos de los códigos de Huffman.

Metodología utilizada

El estudio realizado se fundamentó en la evaluación del rendimiento del algoritmo estático de Huffman (la versión original propuesta en 1952), midiendo el efecto de un conjunto de variables independientes sobre unas variables dependientes, definidas a continuación:

Definición de las variables independientes

Para el presente estudio se definieron las siguientes variables independientes con sus respectivos valores:

1. Tamaño del alfabeto: representa la cantidad de símbolos del alfabeto ASCII que se utilizaron para construir el archivo. Los valores seleccionados fueron: pequeño = 0 (entre 1 y 10 caracteres), mediano = 1 (entre 20 y 30 caracteres) y grande = 2 (entre 40 y 50 caracteres). Es importante resaltar que para lograr la igualdad entre los tamaños de los archivos con aparición equiprobable y aleatoria de sím-

bolos, se garantizó que los valores pequeño, mediano y grande correspondientes al tamaño del alfabeto fuesen múltiplos, y además que el valor del tamaño del alfabeto grande (2) fuese el mínimo común múltiplo de los valores. Es importante resaltar que, en la mayoría de los casos de la vida real, los alfabetos están compuestos por un número de símbolos que es igual a una potencia de 2. Sin embargo, se consideró importante estudiar el comportamiento del algoritmo bajo la variación de este parámetro como mecanismo para analizar su funcionamiento.

2. Entropía: representa el comportamiento probabilístico de la aparición de los símbolos en el archivo. Los valores seleccionados fueron equiprobables = 0 (cuando los símbolos del alfabeto aparecen con la misma frecuencia en el archivo) y aleatorio = 1 (cuando los símbolos del alfabeto aparecen con distinta frecuencia en el archivo).

3. Tamaño del archivo: representa la cantidad de símbolos del alfabeto ASCII que se utilizaron para construir el archivo. Los valores seleccionados fueron: pequeño= 0 (entre 10000 y 99999 caracteres), mediano= 1 (entre 100000 y 999999 caracteres) y grande= 2 (entre 1000000 y 9999999 caracteres). Para lograr la igualdad del tamaño de los archivos con alfabetos aleatorios y equiprobables, se restó a los valores aleatorios de cada rango del tamaño del archivo el resto de la división del valor aleatorio entre el tamaño del rango grande del tamaño del alfabeto (mínimo común múltiplo).

La selección de los niveles de las variables independientes antes definidas se fundamentó en un muestreo no aleatorizado, producto del consenso entre los autores y los expertos en el área estadística.

Definición de las variables dependientes

Para el presente estudio se definieron como variables dependientes los siguientes parámetros, producto de la ejecución del algoritmo de Huffman:

1. Relación de compresión: métrica que permite determinar el grado de minimización en la representación de un archivo.

2. Tiempo de compresión: es la cantidad de unidades de tiempo que se tarda el algoritmo de Huffman implementado en realizar el proceso de codificación de un archivo.

3. Tiempo de descompresión: es la cantidad de unidades de tiempo que se tarda el algoritmo de Huffman implementado en realizar el proceso de decodificación de un archivo.

Determinación del número de repeticiones a realizar por experimento

Para garantizar la validez del análisis estadístico de los datos y considerando los grados de libertad de las variables independientes tamaño del archivo, entropía y tamaño del alfabeto, y sus interacciones, se determinó que eran necesarias como mínimo tres repeticiones.

Generación de los archivos de prueba

Conociendo el número de repeticiones, se procedió a generar los 54 archivos producto de la iteración de las variables independientes (3 niveles del tamaño del alfabeto, 3 niveles del tamaño del archivo y 2 niveles de entropía) con la finalidad de aplicar a cada uno de estos el algoritmo de Huffman para obtener los valores de las variables dependientes.

Ejecución del algoritmo de Huffman

Generados los 54 archivos de prueba, se procedió a implementar el algoritmo de Huffman en un lenguaje de alto nivel (C++) en una plataforma de *hardware* compuesta por un computador de escritorio, con procesador Pentium IV de 2.4 GHz, 1 GB de memoria y 80 GB de disco duro, para luego aplicar el algoritmo a estos archivos. Como resultado de esta aplicación, se obtuvieron los valores de las variables dependientes necesarios para poderlos analizar estadística-

mente. Es importante resaltar que la plataforma de *hardware* utilizada fue la misma en todas las ejecuciones del algoritmo (54) razón por la cual la plataforma de hardware afectó los resultados obtenidos en las pruebas.

Análisis estadístico de los resultados

El diseño seleccionado para el estudio de las variables dependientes, relación de compresión (RC), tiempo de compresión (TCOMP) y tiempo de descompresión (TDCOMP), fue un bloque al azar utilizando el tamaño del alfabeto como factor de bloque y los tratamientos en un arreglo factorial 3x2, es decir, un arreglo factorial de dos factores: uno con tres niveles y otro con dos y tres repeticiones. Para el análisis de los datos se utilizó el procedimiento ANOVA del paquete estadístico SAS (Statistical Analysis System). El modelo aditivo lineal que representa el comportamiento de cada una de las variables en estudio es el siguiente:

$$Y_{ijkl} = \mu + A_i + B_j + C_k + (AB)_{ij} + (AC)_{ik} + (BC)_{jk} + (ABC)_{ijk} + E_{ijkl}$$

Donde

$i = 1, 2, 3$

$j = 1, 2$

$K = 1, 2, 3$

$L = 1, 2, 3$

$\mu =$ la media de la variable en estudio

$A_i =$ efecto del i -ésimo tamaño del alfabeto

$B_j =$ efecto del j -ésimo tamaño del archivo

$C_k =$ efecto del k -ésimo nivel de entropía

$(AB)_{ij} =$ efecto de la interacción entre el i -ésimo tamaño del alfabeto y el j -ésimo tamaño del archivo

$(AC)_{ik} =$ efecto de la interacción entre el i -ésimo tamaño del alfabeto y el k -ésimo nivel de entropía

$(BC)_{jk} =$ efecto del j -ésimo tamaño del archivo y el k -ésimo nivel de entropía

$(ABC)_{ijk}$ = efecto de la interacción entre el i-ésimo tamaño del alfabeto, el j-ésimo tamaño del archivo y el k-ésimo nivel de entropía

E_{ijkl} = error experimental

Resultados obtenidos

Después de ejecutar el algoritmo de Huffman en los 54 archivos de prueba generados, se obtuvieron los siguientes valores de las variables dependientes:

Para tamaño del alfabeto = 0 (tabla 1).

Para tamaño del alfabeto = 1 (tabla 2).

Para tamaño del alfabeto = 2 (tabla 3).

Análisis y discusión de los resultados

Variable relación de compresión (RC)

El análisis de la varianza para la variable relación de compresión (RC) indica diferencias significativas ($P \leq 0,01$) para el tamaño del alfabeto, para el tamaño del archivo, para la entropía y para todas las interacciones analizadas. Todo esto indica que la variable relación de compresión es afectada en su comportamiento tanto por el tamaño del alfabeto como por el tamaño del archivo y la entropía, y que estos factores dependen unos de otros en su acción sobre la variable en cuestión.

Dado el efecto significativo de estos factores sobre la variable relación de compresión

Tabla 1
Tamaño del alfabeto entre 1 y 10 caracteres

Archivo	Entropía	Repetición	Relación de compresión (%)	Tiempo de compresión (s)	Tiempo de descompresión (s)
0	0	0	79,1427537	0,3846154	0,2747253
0	0	1	79,1427537	0,3296703	0,2747253
0	0	2	79,1427537	0,3296703	0,2747253
0	1	0	79,16085	0,3296703	0,2747253
0	1	1	79,1789463	0,3296703	0,2747253
0	1	2	79,1841166	0,3296703	0,2747253
1	0	0	79,1639213	3,1868132	2,8571429
1	0	1	79,1639213	3,2417582	2,8021978
1	0	2	79,1639213	3,1868132	2,8021978
1	1	0	79,1646964	3,2417582	2,8571429
1	1	1	79,1648256	3,4615385	2,8021978
1	1	2	79,165084	3,4615385	2,8021978
2	0	0	79,1663889	31,8681319	28,021978
2	0	1	79,1663889	31,6483516	27,967033
2	0	2	79,1663889	31,8131868	27,9120879
2	1	0	79,1664793	32,1428571	28,2417582
2	1	1	79,1664277	32,1428571	28,2417582
2	1	2	79,1664277	32,1428571	28,1868132

Fuente: elaboración propia.

Tabla 2
Tamaño del alfabeto entre 20 y 30 caracteres

Archivo	Entropía	Repetición	Relación de compresión (%)	Tiempo de compresión (s)	Tiempo de descompresión (s)
0	0	0	43,9067266	0,7692308	0,7142857
0	0	1	43,9067266	0,7142857	0,6593407
0	0	2	43,9067266	0,7692308	0,7142857
0	1	0	43,9610155	0,7142857	0,6593407
0	1	1	43,9881599	0,7692308	0,6593407
0	1	2	44,0101339	0,7142857	0,7142857
1	0	0	44,0336013	6,7582418	6,7032967
1	0	1	44,0336013	6,5384615	6,7582418
1	0	2	44,0336013	6,5384615	6,7032967
1	1	0	44,0513011	6,5934066	6,7582418
1	1	1	44,0496216	6,5934066	6,7582418
1	1	2	44,0489756	6,8131868	6,7582418
2	0	0	44,0459395	65,4945055	67,0879121
2	0	1	44,0459395	67,1428571	66,978022
2	0	2	44,0459395	65,4395604	67,0879121
2	1	0	44,0475932	65,989011	67,4725275
2	1	1	44,0474899	65,8241758	67,3626374
2	1	2	44,0475157	65,8791209	67,5274725

Fuente: elaboración propia.

sión, se realizó la prueba de medias por el método de Duncan, con lo cual pudieron observarse diferencias significativas ($P \leq 0,01$) entre los tres tamaños de alfabeto. Los mayores porcentajes de relación de compresión fueron para el tamaño de alfabeto 0 y los menores para el tamaño 2. Se observaron también diferencias significativas ($P \leq 0,01$) entre los tres tamaños de archivo, con los porcentajes de relación de compresión más bajos para el tamaño 0 y los más altos para el tamaño 2. Con respecto a la entropía, también se pudieron detectar diferencias significativas ($P \leq 0,01$) entre los dos niveles estudiados, con el porcentaje de relación de compresión más alto para el nivel 1.

Con respecto a la interacción tamaño del archivo por entropía, se detectaron diferencias significativas ($P \leq 0,01$) entre la combinación tamaño del archivo 0 con entropía 0 y el resto de las combinaciones, así como también entre la combinación tamaño del archivo 0 y entropía 1 y el resto de las combinaciones, y entre la combinación de tamaño del archivo 1 y entropía 0 y el resto de las combinaciones. Sin embargo, no se detectaron diferencias significativas entre las combinaciones tamaño del archivo 1 y entropía 1, tamaño del archivo 2 y entropía 0 y tamaño del archivo 2 y entropía 1, ni tampoco entre las combinaciones tamaño del archivo 2 y entropía 0 y tamaño de archivo 2 y entropía 1. El más alto porcentaje de relación

Tabla 3
Tamaño del alfabeto entre 40 y 50 caracteres

Archivo	Entropía	Repetición	Relación de compresión (%)	Tiempo de compresión (s)	Tiempo de descompresión (s)
0	0	0	31,2716509	0,7692308	0,8241758
0	0	1	31,2716509	0,989011	0,8241758
0	0	2	31,2716509	0,8791209	0,8791209
0	1	0	31,3698878	0,7692308	0,8791209
0	1	1	31,3931544	0,989011	0,7692308
0	1	2	31,3479138	0,8791209	0,8241758
1	0	0	31,520068	7,6373626	8,1318681
1	0	1	31,520068	7,8571429	8,0769231
1	0	2	31,520068	7,8021978	8,0769231
1	1	0	31,5487495	7,8571429	8,1318681
1	1	1	31,5528838	7,8571429	8,1318681
1	1	2	31,5462948	7,8571429	8,1318681
2	0	0	31,5443182	76,5934066	81,043956
2	0	1	31,5443182	76,4835165	80,989011
2	0	2	31,5443182	76,7032967	80,989011
2	1	0	31,5471733	77,8021978	81,3736264
2	1	1	31,5473154	76,7032967	81,3736264
2	1	2	31,5471475	76,5384615	81,4285714

Fuente: elaboración propia.

de compresión se observó para las combinaciones tamaño del archivo 1 y entropía 0, tamaño del archivo 1 y entropía 1, tamaño del archivo 2 y entropía 0 y tamaño del archivo 2 y entropía 1.

Variable tiempo de compresión (TCOMP)

El análisis de la varianza para esta variable muestra diferencias significativas ($P \leq 0,01$) tanto para el tamaño del alfabeto como para el tamaño del archivo y para la interacción entre el tamaño del alfabeto y el tamaño del archivo, mas no se observaron diferencias entre los niveles de entropía, lo cual indica que el tiempo de compresión es afectado por el tamaño del alfabeto y por el tamaño del archivo, factores que actúan de-

pendientemente sobre la variable en cuestión. Al realizar la prueba de medias, se observaron diferencias significativas ($P \leq 0,01$) entre los tres tamaños de alfabeto, detectándose el tiempo de compresión más bajo (11,87 s) para el tamaño 0 y el más alto, para el tamaño 2. El mismo comportamiento se observó en el caso del tamaño del archivo, con el tiempo de compresión promedio más bajo para el tamaño del archivo 0.

En relación con la interacción tamaño del alfabeto por tamaño del archivo, se observaron diferencias significativas entre casi todas las nueve combinaciones; sin embargo, puede decirse que los tiempos de compresión más bajos los muestran las combinaciones de los tres tamaños del alfabeto

con el tamaño del archivo 0, y los más altos se observan con los tres tamaños del alfabeto y el tamaño del archivo 2.

Variable tiempo de descompresión (TDCOMP)

El análisis de la varianza para esta variable muestra diferencias significativas ($P = 0.01$) tanto para el tamaño del alfabeto como para el tamaño del archivo y su interacción, mas no para la entropía, lo cual indica una clara influencia de esos dos factores sobre el comportamiento de la variable tiempo de descompresión. Según la prueba de medias por el método de Duncan, hay diferencias entre los tres tamaños del alfabeto, observándose el menor tiempo de descompresión para el tamaño del alfabeto 0, seguido del tamaño del alfabeto 1 y, finalmente, el tamaño 2. En cuanto al tamaño del archivo, existen diferencias significativas ($P \leq 0,01$) entre los tres tamaños estudiados, con el menor tiempo de descompresión para el tamaño 0.

En relación con la interacción tamaño del alfabeto por tamaño del archivo, se detectan diferencias significativas entre todas las combinaciones, observándose los tiempos de descompresión más bajos para las combinaciones de los tres tamaños de alfabeto con el tamaño del archivo 0 y los más altos, con el tamaño del archivo 2.

Los resultados antes descritos son el fundamento para dar validez a la selección de los niveles de las variables independientes definidas en el estudio, debido a la significancia obtenida, según el modelo estadístico planteado, en relación con el efecto de las variables independientes y sus respectivas interacciones sobre las variables dependientes.

Conclusiones

Como resultado del análisis estadístico de los datos obtenidos mediante la aplicación del algoritmo de compresión de Huffman a los diferentes archivos de texto generados variando los parámetros tamaño

del alfabeto, tamaño del archivo y la entropía, se puede concluir lo siguiente:

1. El tamaño del archivo y el tamaño del alfabeto tienen influencia sobre el comportamiento de la relación de compresión, el tiempo de compresión y el tiempo de descompresión.

2. La entropía sólo afecta el comportamiento de la variable relación de compresión.

3. La variable relación de compresión es afectada en su comportamiento por el tamaño del alfabeto, el tamaño del archivo, la entropía y sus interacciones, con los mayores porcentajes de relación de compresión, en este caso, para el tamaño del alfabeto 0, para el tamaño del archivo 2 y para el nivel 1 de entropía.

4. La variable tiempo de compresión es afectada por el tamaño del alfabeto, el tamaño del archivo y su interacción, más no por la entropía. Se observan los tiempos de compresión más bajos para el tamaño del alfabeto 0 y para el tamaño del archivo 0; y en el caso de la interacción, los tiempos de compresión más bajos son los correspondientes a los tres tamaños del alfabeto combinados con el tamaño del archivo 0.

5. El tiempo de descompresión es afectado por el tamaño del alfabeto, el tamaño del archivo y su interacción. Se observan los tiempos de descompresión más bajos para el tamaño del alfabeto 0 y el tamaño del archivo 0. Con respecto a las interacciones, los tiempos de descompresión más bajos se observan para los tres tamaños del alfabeto en combinación con el tamaño del archivo 0.

El análisis de varianza aplicado determinó que para la relación de compresión, todas las variables independientes (y algunas de sus interacciones) tienen un efecto significativo, mientras que para el tiempo de compresión y el tiempo de descompresión sólo tuvieron un efecto significativo el tamaño del alfabeto y el tamaño del archivo (y algunas de sus interacciones).

En relación con la importancia del estudio del algoritmo de Huffman a pesar de su longevidad, el análisis sobre los estudios relacionados con la aplicación del algoritmo a problemas de compresión de video e imágenes, así como su uso para la enseñanza de los fundamentos de la teoría de la información, permiten aseverar que este algoritmo seguirá siendo utilizado para otras aplicaciones de compresión de datos por lo que el estudio de su funcionamiento resulta relevante.

Referencias bibliográficas

1. SHANNON C.E. *Bell Systems Technical Journal* 27: 379-423, 623-656, 1948.
2. SAYOOD, K. *Introduction to data compression*. Morgan Kaufmann Publishers, INC. San Francisco, California (USA), 1996.
3. HUFFMAN, D. *A method for the construction of minimum redundancy codes*. In Proc. IRE. 40 (9): 1098-1101, 1952.
4. BOOKSTEIN, A., KLEIN, S. T., AND RAITA, T. "Is Huffman coding dead?" (extended abstract). In Korfhage R., Rasmussen E., Willett P. (eds.) *SIGIR '93*. ACM, New York (USA), pp. 80-87, 1993.
5. HAMADA, M. *Web-based tools for active learning in information theory*. In SIGCSE '07. ACM, New York (USA), pp. 60-64, 2007.
6. KEELER, J. *J. Comput. Small Coll.* 19(5): 289-290, 2004.
7. SILVA DE MOURA E., NAVARRO G., ZIVIANI N., BAEZA-YATES, R. *ACM Trans. Inf. Syst.* 18(2): 113-139, 2000.
8. GOPAL LAKHANI. *Modified JPEG Huffman coding*. IEEE Transactions on Image Processing, vol. 12, núm. 2, pp. 159-169, 2003.
9. JAE-SIK LEE, JONG-HOON JEONG, TAE-GYU CHANG. *An efficient method of Huffman decoding for MPEG-2 AAC and its performance analysis*. IEEE Transactions on Speech and Audio Processing, vol. 13, núm. 6, pp. 1206-1209, 2005.
10. CHANG Y.-W., TRUONG T.-K., CHANG Y. *Direct mapping architecture for JPEG Huffman decoder*. IEE Proc. Commun., vol. 153, núm. 3, pp. 333-340, 2006.
11. JIANTAO ZHOU, ZHIQIN LIANG, YAN CHEN, OSCAR C. AU. *Security analysis of multimedia encryption schemes based on multiple Huffman table*. IEEE Signal Processing Letters, vol. 14, núm. 3, pp. 201-204, 2007.