

Comparación de Métricas para Función de Fitness en Estrategias Evolutivas Aplicadas al Problema Inverso de los IFS

Paula Millado¹, María Laura Ivanissevich¹ y Claudio Delrieux²

¹Universidad Nacional de la Patagonia Austral - L. de la Torre 860 - Río Gallegos - ARGENTINA

²Universidad Nacional del Sur - Av. Alem 1253 - Bahía Blanca - ARGENTINA

e-mail:mivanissevich@uarg.unpa.edu.ar - claudio@acm.org

1 Introducción

Los sistemas de funciones iteradas (IFS) constituyen una manera económica y versátil de representar imágenes por medio de conjuntos fractales. Una imagen cualquiera puede ser arbitrariamente aproximada por un atractor fractal adecuadamente representado por un IFS. La búsqueda de dicho conjunto constituye el problema inverso del IFS. Lamentablemente por fuerza bruta esta búsqueda es inabordable. Las estrategias hasta ahora propuestas para solucionar esta dificultad son dos. Por un lado, es posible restringir el espacio de búsqueda a un subconjunto de las transformaciones afines, como se propone en la compresión fractal en bloques (CFB) [3, 6]. Los resultados obtenidos con la CFB poseen una buena relación de compromiso entre tiempo de cómputo, compresión resultante, y calidad final, aunque están evidentemente lejos de las posibilidades teóricas [4].

La otra posibilidad, explorada en nuestra línea de trabajo, es utilizar un mecanismo de búsqueda adaptativo, que tenga cierta inteligencia como para guiarse dentro del un subconjunto del espacio de búsqueda en el cual los resultados se acercan a la solución buscada. Este es el tipo de estrategia buscada con la programación evolutiva (PE) [11] y con la programación genética (PG) [8]. El uso de PE y PG para la solución del problema inverso ha sido mucho menos estudiado, pese a que permitiría obtener mejores compresiones, tal vez con mayores tiempos de cómputo, que la CFB. Sin embargo, dada la naturaleza asimétrica de la compresión fractal, y el hecho de que una imagen comprimida es utilizada innumerables veces, este tiempo adicional representa un costo menor frente a la ventaja de tener imágenes mejor comprimidas.

En este trabajo se muestra la performance de las dos métricas seleccionadas tras un proceso de evaluación sobre un conjunto de métricas aplicables a nuestro problema específico [5]. Es importante notar que el problema inverso representa un caso interesante de estudio para proponer y comparar diversas estrategias evolutivas y genéticas, diferentes funciones de adaptación y selección, etc. Para obtener y visualizar estos resultados se implementó un ambiente gráfico que permite testear y comparar los resultados y evaluar numérica y visualmente los resultados que se van produciendo a lo largo de las generaciones.

2 Programación evolutiva aplicada al problema inverso de los IFS

Un sistema de funciones iteradas (IFS) consiste en una colección de transformaciones afines contractivas que mapea al plano \mathfrak{R}^2 sobre sí mismo. Esta colección de transformaciones define un mapa $W(\cdot) = \bigcup_{i=1}^n w_i(\cdot)$. Estas tienen la importante característica de que cuando son aplicadas repetidamente, convergen a un punto fijo. Dado un conjunto inicial S (colección de puntos del plano), podemos calcular $w_i(S)$ para cada i , tomar la unión de estos conjuntos y obtener un nuevo conjunto $W(S)$. Según el teorema de Hutchinson [1], cuando las transformaciones w_i son contractivas, entonces W es contractiva en el espacio (cerrado y acotado) de subconjuntos del plano, y la iteración de un IFS converge a un único subconjunto del plano, el cual es un punto fijo del mapa W , denominado *atractor* A del IFS.

La obtención del código IFS de una figura I arbitraria, se realiza partiendo de una representación inicial aproximada de la silueta S del objeto que se desea modelar. La idea del método es buscar un grupo de mosaicos —copias transformadas de S — que en conjunto cubran la silueta original (Teorema del Collage [2]). Si por $d(I, J)$ denotamos la distancia de Hausdorff entre subconjuntos I y J del plano en una métrica adecuada, y r es la contractividad global de W , entonces el teorema del collage permite afirmar la desigualdad

$$d(A, I) \leq \frac{1}{1-r} d(I, \bigcup w_i(I)),$$

lo cual significa que la distancia entre la figura I y el atractor A del IFS está acotada por la distancia entre I y el cubrimiento determinado por W . A mejores cubrimientos, mejor aproximación a la imagen final. En otras palabras, para lograr la codificación se deben determinar un conjunto de transformaciones afines contractivas $\{w_1, w_2, \dots, w_n\}$ con la siguiente propiedad: La silueta original S y el conjunto $\tilde{S} = \bigcup_{n=1}^N w_n(S)$ deben ser visualmente cercanos, siendo el número de transformaciones el menor posible. $w_n(S)$ es llamado el n -ésimo mosaico del collage. Este cubrimiento hasta ahora no ha podido implementarse en forma automática, debiéndose trabajar con asistencia humana. En este trabajo proponemos el uso de Programación Evolutiva (PE) y Programación Genética (PG) como mecanismo para resolver el problema inverso de los IFS.

La (PE) y la (PG) son técnicas bien conocidas que surgen de imitar lo que conocemos de la evolución natural. Las ideas principales provienen de una metáfora de la evolución natural, en la cual existen individuos (fenotipos) que expresan una información genética (genotipo), y además están sujetos a la presión evolutiva del medio (fitness). Los individuos más exitosos (con mejor fitness) ven aumentadas sus posibilidades reproductivas, generando nuevos individuos con su mismo genotipo. Sin embargo, en el proceso reproductivo ocurre el fenómeno distintivo de la evolución natural: la aparición de mutaciones al azar. Este mismo esquema dinámico, con un sinnúmero de variaciones, es especial para resolver problemas que con los métodos de Inteligencia Artificial de alto nivel tradicionales (búsqueda heurística, lógica, etc.) son o bien intratables o bien imprácticos.

Supongamos la existencia de una población con μ ancestros, cuyo genotipo \bar{g}_i se encuentra codificado con números. Entonces se crea un conjunto λ de descendientes siguiendo el esquema

$$\bar{p}_i = (\bar{g}_i, \bar{\sigma}_i) \xrightarrow{\text{mut}} (\bar{g}_i + n_0(\bar{\sigma}_i), \alpha(\bar{\sigma}_i)),$$

donde $n_0(\bar{\sigma}_i)$ denota la elección de un número de una aleatoria normal con media cero y variancia σ_i . α define una función de adaptación, la cual puede ser la distancia de Hamming

o la distancia de Hausdorff entre el atractor A del IFS resultante y la imagen I de entrada. En nuestro caso, \bar{g}_i representa los parámetros de la IFS W que estamos tratando de encontrar. Cada mapa w_k requiere seis reales para su objetivo:

$$w_k(x, y) = \begin{pmatrix} a_k & b_k \\ c_k & d_k \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e_k \\ f_k \end{pmatrix} \quad (1)$$

$$= \begin{pmatrix} r_{1k} \cos(\theta_{1k}) & -r_{2k} \sin(\theta_{1k}) \\ r_{1k} \sin(\theta_{2k}) & r_{2k} \cos(\theta_{2k}) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e_k \\ f_k \end{pmatrix} \quad (2)$$

En general, la expresión 2 es preferible a la 1 por tener un significado geométrico más intuitivo. Por lo tanto, \bar{g}_i representará las n sextuplas $(r_{1k}, r_{2k}, \theta_{1k}, \theta_{2k}, e_k, f_k), k \in [1..n]$ en un determinado individuo de una generación. Como estrategia híbrida se puede introducir, además de la mutación, el *crossover* o cruzamiento entre el genotipo de dos ancestros para obtener el genotipo del sucesor. Para evitar los mínimos locales, y quedar atrapado en un “nicho ecológico”, la estrategia mantiene una diversidad genotípica seleccionando un grupo numeroso de individuos con la mejor adaptación. Entre estas técnicas de *niching* podemos contar el *fitness sharing* (escalar el fitness en un entorno de cada individuo), el *crowding* (reemplazar al padre más cercano del individuo recientemente creado), y el *clustering* o agrupamiento, la cual resulta ser la más eficiente en la práctica.

En trabajos anteriores se implementó la función de adaptación por medio de la función de Hamming [9, 7]. En este trabajo veremos que el uso de la función de Hausdorff, si bien computacionalmente más compleja, produce resultados geoméricamente más adecuados. La distancia de Hausdorff es una métrica bien conocida para el espacio de los compactos de R^2 [2]. Se obtiene el espacio métrico completo de los compactos, $(H(R^2), h)$ definiendo la *distancia forward* y la *distancia backward* del siguiente modo:

$$df(A, B) = \bigvee_{a \in A} \bigwedge_{b \in B} d(a, b)$$

$$db(A, B) = \bigvee_{b \in B} \bigwedge_{a \in A} d(a, b)$$

Para obtener una función simétrica se define la distancia de Hausdorff como:

$$h(A, B) = df(A, B) \bigvee db(A, B) \quad (3)$$

La métrica de Hausdorff puede ser usada como medida de la “cercanía” de dos imágenes apropiadamente configuradas computacionalmente. La expresión 3 es topológicamente equivalente a la expresión:

$$h'(A, B) = df(A, B) + db(A, B) \quad (4)$$

El algoritmo implementado es de tiempo lineal y consiste en computar las matrices binarias de las dos imágenes, una de ellas la imagen objetivo, I_{obj} , y la otra la aproximación que se va generando en el sistema evolutivo, I_{aprox} , através de la construcción de una matriz de distancias, D_b , (y la D_f), conteniendo la información de la distancia Manhattan que permite obtener el dato $\bigvee_{a \in A} \bigwedge_{b \in B} d(a, b)$ (y el análogo, ver [10]).

$$\begin{aligned} db(I_{obj}, I_{aprox}) &= \max\{D_b[i, j] : I_{aprox}[i, j] = 1\} \\ df(I_{obj}, I_{aprox}) &= \max\{D_f[i, j] : I_{obj}[i, j] = 1\} \\ h(I_{obj}, I_{aprox}) &= db(I_{obj}, I_{aprox}) + df(I_{obj}, I_{aprox}) \end{aligned}$$

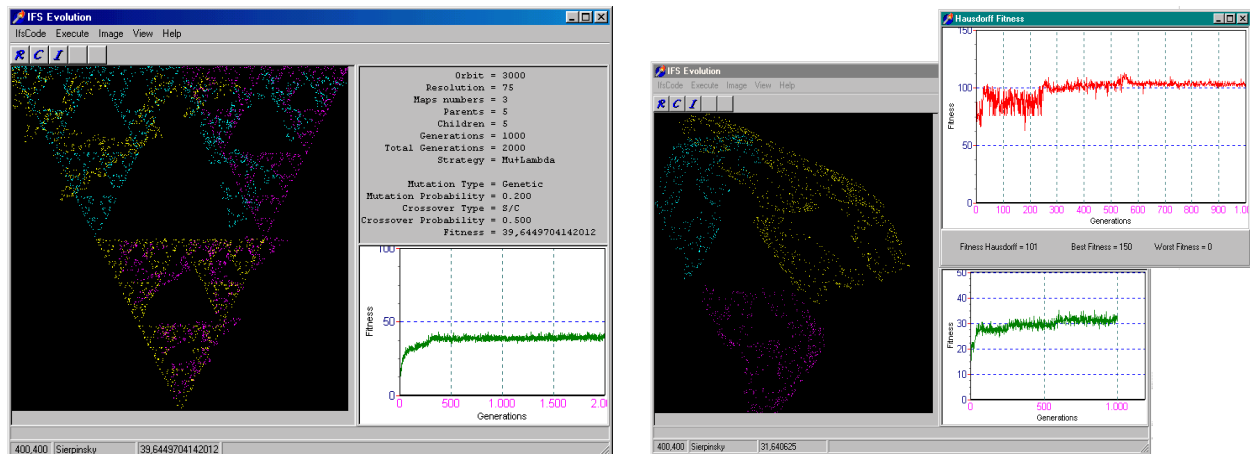


Figura 1: (a) La interfase del sistema, mostrando la convergencia evolutiva de un conjunto de transformaciones al atractor de la imagen inicial, y (b) la representación simultánea de las funciones de fitness con diferentes medidas.

3 Resultados obtenidos

Para contar con una implementación adecuada en velocidad y capacidad de cómputo, el programa fue implementado en el lenguaje Delphi, construyéndose un ambiente gráfico para testear y evaluar las diferentes estrategias y funciones. En la Fig. 1(a) es posible observar la interfase gráfica del sistema. Abajo a la izquierda se muestra el atractor resultante, y abajo a la derecha, la curva que representa la evolución del *fitness* del atractor encontrado a lo largo de las generaciones. Los controles de la parte superior izquierda permiten elegir un conjunto de imágenes de entrada, para testear el correcto funcionamiento en varios casos, mientras que los controles a la derecha permiten controlar la ejecución del sistema, modificar los parámetros, las diversas estrategias, la función de fitness, etc. Para poder comparar los resultados de convergencia con las distintas métricas, el programa permite visualizar simultáneamente ambas funciones de fitness (ver Fig. 1(b)).

En la Fig. 2 se muestran los resultados obtenidos al utilizar la distancia de Hausdorff como función de fitness. Aunque el costo en tiempo de computación llega casi a triplicarse, los resultados obtenidos son mucho más adecuados en la convergencia hacia la imagen buscada. Teniendo en cuenta que la decodificación de imágenes por IFS de una u otra calidad insume el mismo tiempo, y que la codificación se hace una sola vez para muchos usos, es entonces importante poder codificar las imágenes con mayor calidad, aunque el tiempo sea mayor.

Referencias

- [1] M. Barnsley, A. Jacquin, y F. Malassenet. Harnessing Chaos for Image Synthesis. *ACM Computer Graphics*, 22(3):131–14, 1988.
- [2] M. F. Barnsley. *Fractals Everywhere*. Academic Press, Boston, 1988.
- [3] Yuval Fisher, editor. *Fractal Image Compression: Theory and Application*. Springer-Verlag, New York, NY, USA, 1995.

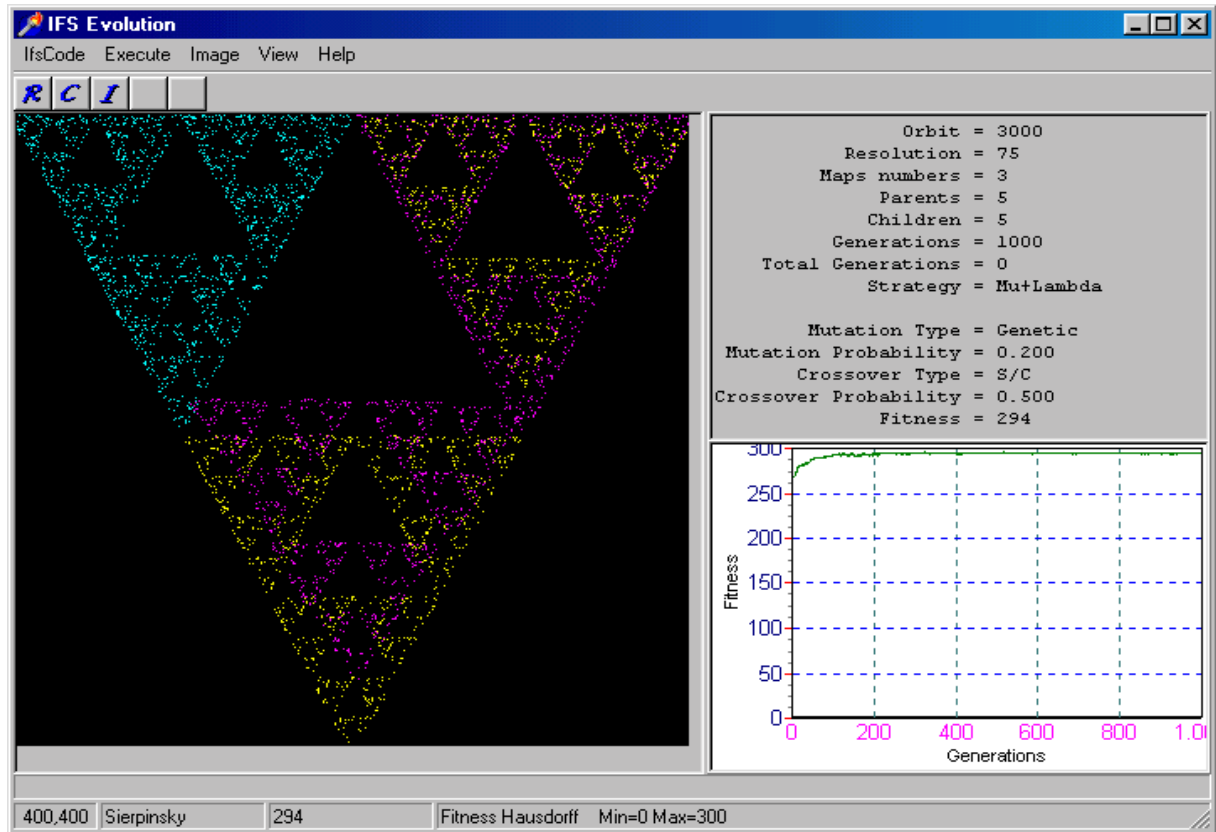


Figura 2: La convergencia con la distancia de Hausdorff es más lenta, pero el atractor es de mucha mejor calidad.

- [4] Yuval Fisher, editor. *Fractal Image Encoding and Analysis*, volumen 159, *NATO ASI Series. Series F: Computer and Systems Sciences*. Springer-Verlag, New York, NY, USA, 1997.
- [5] J. Gutiérrez, A. Cofiño, y M. Ivanishevich. An Hybrid Evolutive-Genetic Strategy for the Inverse Fractal Problem of IFS Models. *Lecture Notes in Artificial Intelligence*, 1952:467–476, 2000.
- [6] Arnaud E. Jacquin. Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformations. En Murat Kunt, editor, *Visual Communications and Image Processing '90*, volumen 1360, *SPIE Proceedings*, págs. 227–239, Lausanne, Switzerland, October 1990.
- [7] R. Klette y P. Zamperoni. Measures of Correspondence Between Binary Patterns. *Image and Vision Computing*, 5:287–295, 1987.
- [8] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, segunda edición, 1994.
- [9] Paula Millado, Luis Sierpe, Laura Ivanishevich, y Claudio Delrieux. Testing Genetic and Evolutive Strategies in the Inverse IFS Problem. En *Proceedings of the CISST 2002 Conference*, págs. 267–274, CSREA Press, ISBN 1-892512-94-7, 2002.
- [10] Shonkwiler. R. An Algorithm for Computing the Hausdorff Distance Efficiently. *Information Processing Letters*, 30:87–89, 1989.
- [11] I. Rechenberg. *Evolution strategie: Optimierung technischer systeme nach prinzipien der biologischen evolution*. Frommann-Holzboog, 1973.