



# OPTIMIZACIÓN DE LA CONECTIVIDAD DE REDES DE MAPAS ACOPLADOS MEDIANTE ALGORITMO GENÉTICO

Ricardo Estévez Basanta

Requisito parcial para la obtención del grado de  
LICENCIADO EN FÍSICA

UNIVERSIDAD DE LOS ANDES  
DEPARTAMENTO DE FÍSICA  
MÉRIDA, VENEZUELA

June 2008

# Índice general

<b>Índice general</b>	<b>III</b>
<b>Resumen</b>	<b>IV</b>
<b>Introducción</b>	<b>1</b>
<b>1. Teoría</b>	<b>3</b>
1.1. Grafos . . . . .	3
1.1.1. Conceptos Básicos de los Grafos . . . . .	4
1.1.2. Matriz de adyacencia . . . . .	6
1.1.3. Caracterización de grafos . . . . .	7
1.1.4. Tipos de Grafos . . . . .	9
1.2. Algoritmo genético . . . . .	16
1.2.1. El lenguaje de los algoritmos genéticos . . . . .	17
1.2.2. Tipos de representación . . . . .	19
1.2.3. Población . . . . .	20
1.2.4. Función objetivo . . . . .	20
1.2.5. Operadores básicos de un algoritmo genético . . . . .	22
1.2.6. Parámetros de un algoritmo genético . . . . .	27
1.3. Mapas unidimensionales . . . . .	28
1.3.1. Puntos fijos y estabilidad local . . . . .	28
1.3.2. Órbitas . . . . .	30
1.3.3. Intermitencia . . . . .	31
1.3.4. Sensibilidad a las condiciones iniciales y exponente de Liapunov . . . . .	34
1.4. Redes de mapas acoplados . . . . .	35
1.4.1. Sincronización . . . . .	36
<b>2. Metodología y resultados.</b>	<b>42</b>
2.1. Metodología . . . . .	42
2.2. Resultados . . . . .	47

<b>3. Discusión de Resultados y Conclusiones.</b>	<b>53</b>
3.1. Discusión de Resultados . . . . .	53
3.2. Conclusiones . . . . .	54
<b>A. Algoritmo Genético del Trabajo</b>	<b>55</b>
<b>Referencias</b>	<b>72</b>

# Resumen

La topología de interconexión de los sistemas complejos determina, en buena medida, el comportamiento macroscópico de los mismos. Este hecho hace pensar que existe algún tipo de red cuya conectividad puede maximizar o minimizar el valor de ciertos parámetros de orden del sistema. Para estudiar esta posibilidad se modifican las características del grafo de interconexión de la red mediante un algoritmo genético que, partiendo de una población inicial de redes de mapas acoplados, crea nuevas redes. En particular, se emplean redes de mapas logarítmicos acoplados con dinámica caótica de sus elementos y el parámetro de orden a minimizar es una función que involucra la desviación estándar y el número de enlaces promedio de los estados de los mapas. Para implementar el algoritmo genético se creó un operador genético que permite combinar matrices de acoplamiento para obtener una nueva generación de redes. Una vez que los mejores individuos de la población converjan, se caracterizan buscando propiedades topológicas comunes. Palabras clave: **Algoritmo Genético, Redes de Mapas Acolados, Optimización, Grafos.**

Este trabajo fue financiado en parte por el Consejo de Desarrollo Científico, Humanístico y Tecnológico de la Universidad de los Andes en Mérida, Venezuela, bajo la subención I-886-05-02-A. Gracias a mi familia y a mi esposa que siempre me han dado su apoyo incondicional. Agradezco por último a mi tutor el Dr. Kay Tucci sinceramente por el apoyo y la ayuda prestados a lo largo de la consecución de este trabajo.

# Introducción

En los últimos tiempos ha aumentado el interés por estudiar sistemas complejos y en especial la relación que existe entre topología y dinámica de la red. Se ha determinado que la estructura de la red juega un rol importante en muchos casos, como por ejemplo en redes sociales, ecosistemas, redes neuronales, el cerebro, Internet o el World Wide Web. Para estudiar este tipo de sistemas se han construido modelos donde, en general, cada nodo de una red sufre un proceso dinámico mientras está acoplado a otros nodos. El comportamiento colectivo del sistema depende fuertemente en la eficiencia de los caminos que comunican los nodos, la cual está dictada por la topología de la red [1, 2, 3].

Particularmente, un gran número de estudios se han enfocado en la sincronización de redes de mapas acoplados, estos últimos introducidos simultáneamente por K. Kaneko, R. Kapral y S. Kuznetsov entre 1983 y 1984 [4, 5, 6, 7, 8, 9]. Sin embargo, la mayoría de estos estudios se han hecho con una conectividad regular, es decir cuando las unidades están acopladas sobre una red cuadriculada a sus vecinos más cercanos o cuando están acopladas globalmente. Últimamente se ha estudiado la sincronización con otros tipos de topología, como son las redes aleatorias [10], las tipo mundo pequeño [11, 12, 13, 14], árboles [15], fractales [16, 17, 18], redes libres de escala [19, 20], entre otras.

En este trabajo se busca determinar las propiedades topológicas de la red que faciliten el surgimiento del fenómeno de sincronización con el mínimo número de enlaces posible. Para ello, partiendo de una población inicial de redes de mapas caóticos acoplados se utiliza un algoritmo genético para crear nuevas redes. Las redes se evalúan mediante una función que involucra la desviación estándar de los estados de los nodos y el número de enlaces promedio de la red. La desviación estándar es el parámetro de orden que refleja el estado de

sincronización del sistema. El algoritmo genético aplica un operador genético a las matrices de acoplamiento diseñado para hacer evolucionar a la población de redes. El proceso se repite tantas veces como sean necesario para que la población converja a la estructura buscada. Por último se analizarán las redes resultantes para encontrar características comunes que reflejen la estructura de las redes obtenidas. Para ello se emplean algunas medidas como el coeficiente de agrupamiento; la longitud característica; el número de enlaces promedio, máximo y mínimo de los nodos de cada red; y la distribución de grados de la red.

En el capítulo 1, se hace una recopilación teórica de las herramientas necesarias para la ejecución del trabajo. La primera parte trata de grafos o redes. Primeramente se habla de conceptos matemáticos básicos así como del sistema de representación mediante la matriz de acoplamiento. Seguidamente explicamos las distintas herramientas para la caracterización de redes entre las que se encuentran el agrupamiento, la longitud característica, el número de enlaces promedio, máximo y mínimo y la distribución de grados de la red. Por último se describen distintos tipos de redes complejas como las redes aleatorias, tipo mundo pequeño, árboles, fractales y regulares. En la segunda parte, se habla de el algoritmo genético. Describimos sus distintas partes como son la población inicial, función objetivo, operadores genéticos y parámetros del algoritmo. En la tercera parte se trata sobre mapas unidimensionales centrandó la explicación en el mapa logarítmico, que es el mapa que se emplea en la simulación. Se habla de diagramas de bifurcación, exponente de Liapunov, puntos fijos, estabilidad local y órbitas entre otras cosas. Finalmente se describen las redes de mapas acoplados y la sincronización.

En el capítulo 2, se describe la metodología utilizada para la elaboración del trabajo. Se especifica, entre otras cosas, la disposición de la población inicial de redes, los métodos de recombinación y mutación utilizados, la aplicación del algoritmo genético y parámetros utilizados como el parámetro de control del mapa unidimensional y el acoplamiento de la red de mapas acoplados. Seguidamente se muestra el progreso generacional de la función de evaluación, las medidas utilizadas para la descripción de las redes obtenidas y un análisis de los resultados.

En las conclusiones se hace un recuento de los pasos utilizados, se analizan los resultados obtenidos y se proponen temas nuevos para la investigación. Al final del trabajo se encuentran los anexos donde se encuentra la notación utilizada en el algoritmo genético.

# Capítulo 1

## Teoría

En el presente capítulo se describen las herramientas necesarias para la elaboración de este trabajo. En la sección 1.1 se introducen conceptos básicos sobre grafos como son matriz de adyacencia, caracterización de grafos y tipos de grafos. En la sección 1.2 se hace una descripción de los algoritmos genéticos haciendo énfasis en los conceptos de población inicial, función objetivo, operadores genéticos y los parámetros del algoritmo. Seguidamente en la sección 1.3 se habla de los mapas unidimensionales enfocándonos en el mapa logarítmico que es el mapa que se utilizó en este trabajo. En ella se describe que son los diagramas de bifurcación, el exponente de Liapunov, los puntos fijos, la estabilidad local y las órbitas. Por último en la sección 1.4 se tratan las redes de mapas acoplados y su sincronización.

### 1.1. Grafos

En nuestra vida diaria nos vemos rodeados por cosas que pueden representarse mediante una abstracción matemática que se denomina grafos y que nos parecen cotidianas como las carreteras, líneas telefónicas, Internet, líneas de transporte público, entre muchas otras.

En esta sección se explica qué son los grafos, sus tipos, como se caracterizan y la representación con matriz de adyacencia.

### 1.1.1. Conceptos Básicos de los Grafos

Los grafos son una abstracción matemática [21, 22, 23]. Se define un grafo  $G$  como un par ordenado de  $V$  y  $E$ , donde  $V$  es un conjunto de vértices o nodos y  $E$  un conjunto de enlaces o aristas. Se denota el grafo como  $G = (V, E)$ .

Para cada enlace, de  $G$ , se asocia una pareja ordenada o no ordenada de vértices que pertenezcan al grafo. Si  $x \in E$  entonces está asociado con un par ordenado o no ordenado de vértices  $(u, v)$ , donde  $u, v \in V$ . La figura 1.1 muestra un grafo. En ella podemos ver que si dos nodos están conectados por la misma arista, por ejemplo los vértices 1 y 3, se llamarán vértices adyacentes. Si una arista está conectada a un vértice serán arista y vértice incidentes. Si dos aristas se conectan al mismo vértice se llaman aristas adyacentes. Si un enlace sale y entra en el mismo vértice se le denomina bucle o lazo. Si dos aristas tienen como extremos los mismos vértices se las denomina aristas múltiples.

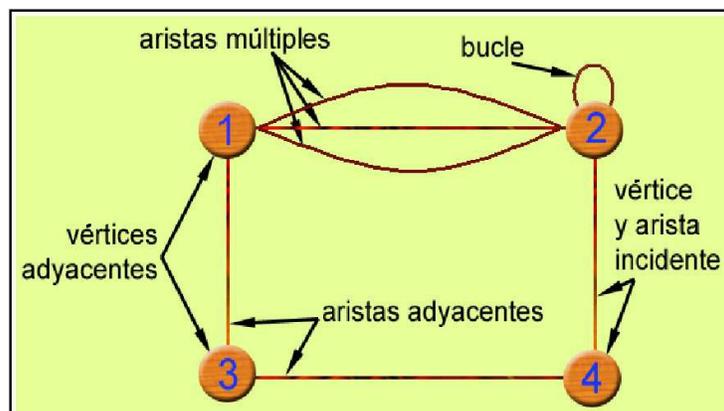


Figura 1.1: Proceso de modelado tradicional.

Un grafo que contenga aristas múltiples se le llama multígrafo, y el que no contenga aristas múltiples ni bucles se le llama grafo simple.

Un enlace de  $G$  asociado a una pareja de nodos ordenados se denomina enlace dirigido y un enlace asociado a un par de nodos no ordenados, enlace no dirigido. Si el enlace carece de dirección se denota indistintamente como  $(u, v)$  o  $(v, u)$ , siendo  $u$  y  $v$  los vértices que enlaza. En un dígrafo o grafo dirigido todos los enlaces de un grafo tienen una dirección, salen de un vértice y terminan en otro. Si todos los enlaces de un grafo son no dirigidos se le denomina grafo no dirigido. Si el grafo contiene enlaces dirigidos y no dirigidos es un

grafo mixto. La figura 1.2 muestra varios ejemplos de grafos de diferente tipo.

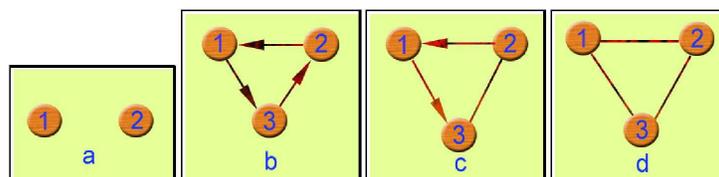


Figura 1.2: El grafo a puede ser considerado dirigido o no dirigido, b es un grafo dirigido, c es mixto y d es no dirigido.

Un grafo con números sobre las aristas, como el mostrado en la figura 1.3, se llama gráfica con pesos. Si la arista tiene la etiqueta  $k$ , se dice que el peso de la arista es  $k$ .

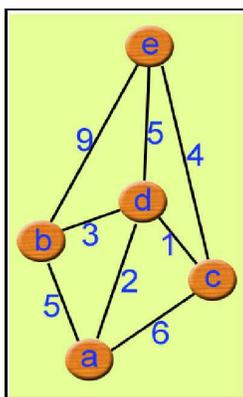


Figura 1.3: Gráfica con pesos.

Un camino en un grafo  $G$ , es una sucesión de  $(v_0, v_1, v_2, \dots, v_n) \in V$  de forma que todos los elementos de la sucesión son vértices y que exista una arista entre todo par de vértices consecutivos en la sucesión.

La longitud de un camino con extremos  $v_0$  y  $v_n$  viene dada por el número de aristas que lo forman si las aristas no tienen peso; de lo contrario, viene dado por la suma de los pesos de las aristas que lo forman.

Un camino es propio si lo forman dos o más vértices; es abierto si sus extremos no coinciden, es decir,  $v_0 \neq v_n$ ; es cerrado si sus extremos coinciden,  $v_0 = v_n$ ; es simple si no se repiten aristas; es elemental si no se repiten vértices, con la excepción quizás de los extremos. En la tabla 1.1 se pueden ver distintos caminos de la figura 1.3 con sus longitudes.

Un Grafo  $G$  es conexo si existe un camino entre cualquier par de sus vértices. Un ciclo

Tipo	Camino	Longitud
Abierto	b,d,c,e,d,a	15
Cerrado	b,a,c,d,e,d,b	25
Simple	b.a.c.d,e,c	21
Elemental	b,a,c,d,e,b	26

tabla 1.1: Tipos de camino con sus longitudes de la gráfica de la figura 1.3.

es un camino de longitud distinta de cero en el que no hay aristas repetidas de  $v$  a  $v$ . Un ciclo simple es un ciclo sin vértices repetidos.

### 1.1.2. Matriz de adyacencia

Esta es la forma más común de representación de grafos [21, 37, 23]. En ella se utiliza una matriz  $n \times n$  donde  $n$  es el número de vértices que hay en el grafo. La matriz  $M[i][j]$  tiene 1 si existe una arista que conecte el vértice  $i$  con el vértice  $j$ , y 0 si no están conectados. En el caso de que el grafo tenga pesos se puede poner el peso de la arista en el lugar en que se encontraba el 1. En la figura 1.4 se puede ver un grafo y su matriz de adyacencia.

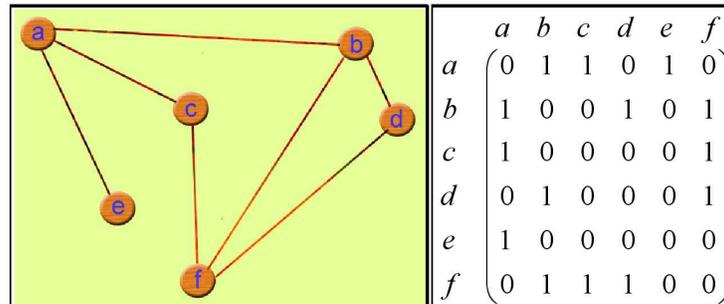


Figura 1.4: Grafo y su matriz de adyacencia.

Se puede obtener el grado de un vértice  $v$   $k_v$  al sumar los unos en el reglón que le corresponde de la matriz de adyacencia. Por ejemplo, en la figura 1.4 el vértice  $a$  tiene tres unos, que se encuentran en el primer reglón de la matriz correspondiente, por lo tanto  $\delta(a) = 3$ . Este tipo de matriz puede representar bucles poniendo un 1 en  $M[i][i]$ . También puede representar aristas paralelas con la modificación propia del grafo con pesos, sólo que, en vez de representar pesos, representa el número de aristas con vértices comunes.

Si  $M$  es la matriz de adyacencia, las potencias de  $M$ ,  $M^n$ , cuentan, en  $M[i][j]$ , el número de caminos de  $i$  a  $j$  de longitud  $n$ . Por ejemplo, si  $M$  es la matriz de adyacencia de la figura 1.4, entonces:

$$M^2 = \begin{matrix} & a & b & c & d & e & f \\ \begin{matrix} a \\ b \\ c \\ d \\ e \\ f \end{matrix} & \begin{pmatrix} 3 & 0 & 0 & 1 & 0 & 2 \\ 0 & 3 & 2 & 1 & 1 & 1 \\ 0 & 2 & 2 & 1 & 1 & 0 \\ 1 & 1 & 1 & 2 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 2 & 1 & 0 & 1 & 0 & 3 \end{pmatrix} \end{matrix}, \quad (1.1)$$

Como se ve en la Ec.(1.1),  $M[b][c]$  dice que hay 2 caminos de longitud 2 que van de  $b$  a  $c$ . Uno de ellos es  $b, f, c$  y el otro  $(b, a, c)$ . Los números en la diagonal de  $M^2$  representan el grado de cada vértice. Como se ve el grado de  $a$  es 3, el de  $b$  es 3, el de  $c$  es 2, etc.

### 1.1.3. Caracterización de grafos

Cada grafo o red tiene rasgos topológicos que caracterizan su conectividad y que influyen en los procesos realizados en ellas. El análisis en las redes depende entonces de las medidas que se puedan realizar que reflejen la estructura de la topología de los grafos [24, 25, 26, 27]. En esta subsección se explica algunas de las medidas que se pueden hacer en las redes para su caracterización.

- Coeficiente de agrupamiento (clustering). En este trabajo se usa el coeficiente de agrupamiento para grafos no dirigidos. Este coeficiente puede usarse para caracterizar la presencia de ciclos simples de orden tres. El número de ciclos simples de grado tres dentro de la red se denomina  $N_{\Delta}$ . Así se tiene que:

$$N_{\Delta} = \sum_{k>j>i} M_{ij}M_{ik}M_{jk}, \quad (1.2)$$

donde  $M_{ij}$  es un elemento de la matriz de adyacencia y la sumatoria es sobre cada grupo de tres vértices distintos  $i, j$  y  $k$  solamente una vez.

Un triple conexo es un grupo de tres vértices donde hay un vértice central que se conecta a los otros dos directamente por medio de un enlace. El número de triples conexos dentro de una red se denomina  $N_3$ , así:

$$N_3 = \sum_{k>j>i} (M_{ij}M_{ik} + M_{ji}M_{jk} + M_{ki}M_{kj}) , \quad (1.3)$$

donde la suma es sobre grupos de vértices diferentes  $i$ ,  $j$  y  $k$  solamente una vez.

El coeficiente de agrupamiento se define entonces como:

$$C = \frac{3N_{\Delta}}{N_3} . \quad (1.4)$$

El número 3 se debe a que un triángulo puede ser visto como tres triples con diferente vértice central.

- Longitud característica. Para un grafo no dirigido y sin pesos el número de vértices en un camino que conecte los vértices  $i$  y  $j$  se le llama longitud del camino. El camino más corto se dice del que, conectando los vértices  $i$  y  $j$ , tiene menor longitud. La longitud del camino más corto o distancia geodésica se le denomina  $d_{ij}$ . Si no existiesen caminos entre los vértices  $i$  y  $j$  entonces  $d_{ij} = \infty$ . La longitud característica es el valor medio de la distancia geodésica:

$$l = \frac{1}{N(N-1)} \sum_{i \neq j} d_{ij} , \quad (1.5)$$

donde la sumatoria es sobre vértices conexos y  $N$  el número de vértices en la red.

- Grado de un vértice. El grado de un vértice  $i$ ,  $k_i$ , es el número de enlaces incidentes a  $i$ . Cuando un vértice tiene un bucle se cuentan dos conexiones como se puede ver en la figura 1.5. El número de enlaces que posee el vértice de mayor grado dentro de un grafo se denomina  $k_{max}$  y el número de enlaces que posee el vértice de menor grado en toda la red se denomina  $k_{min}$ . El promedio de los grados de todos los vértices se denomina

$$\bar{k} = \sum_{i=1}^N \frac{k_i}{N}. \quad (1.6)$$

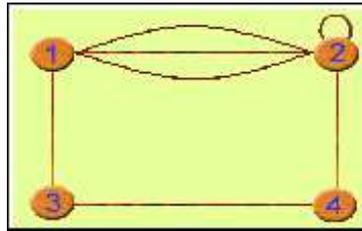


Figura 1.5: Tenemos los siguientes grados:  $k_1 = 4$ ,  $k_2 = 6$ ,  $k_3 = 2$ ,  $k_4 = 2$ .

La suma de los grados de los vértices de una gráfica es par. El teorema de Euler nos dice que si  $G$  es una gráfica con  $n$  aristas y  $v_1, v_2, v_3, \dots, v_n$  vértices, entonces:

$$\sum_{i=1}^N k_i = 2m. \quad (1.7)$$

La distribución de grado,  $P(k)$ , indica la fracción de vértices de una red con grado  $k$ . Dependiendo del tipo de red la distribución puede ser de Poisson, exponencial, función delta y ley de potencias entre otros.

- *Grupo y familia.* Un grupo es un subgrafo  $G'_i = (V'_i, E'_i)$  de  $G = (V, E)$ , con  $V'_i \in V$  y  $E'_i \in E$ , conexo pero aislado de los vértices que no pertenecen al grupo  $V'_i$ . Cada grupo de un grafo  $G$  posee un número de vértices  $V'_i$ , que se llama familia, de forma tal que  $\sum_{i=1}^n V'_i = N$ , con  $n$  el número de grupos y  $N$  el número total de vértices de la red.

#### 1.1.4. Tipos de Grafos

Muchos modelos de redes se han propuesto con la intención de estudiar las propiedades topológicas de redes reales. Algunos de estos modelos han suscitado un gran interés como las redes aleatorias, tipo mundo pequeño, fractales, árboles y regulares entre otras. En esta subsección se describen algunos de estos modelos.

- Grafo aleatorio. Es una red en la cual los enlaces entre los nodos se distribuyen de forma aleatoria. Existen varios métodos para crear redes aleatorias. El más común es el creado por Erdős y Rényi [28, 29, 30] en 1959. En este modelo los  $[n(n-1)]/2$  enlaces potenciales de un grafo no dirigido simple con  $n$  vértices se incluyen independientemente con probabilidad  $0 < p < 1$ . En el proceso se asigna a cada enlace potencial un número  $r \in [0, 1)$  uniforme aleatorio, y se inserta el enlace si  $r < p$ . La distribución de grado del modelo de Erdős y Rényi tiene forma de distribución de Poisson, como se ve en la figura 1.6.

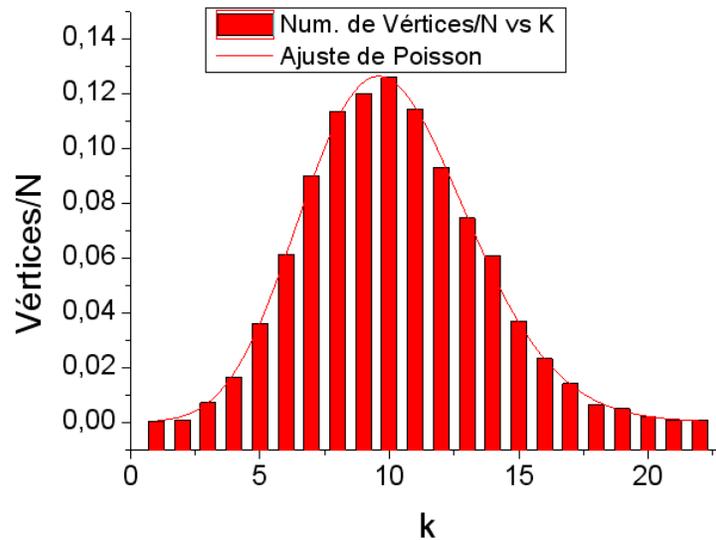


Figura 1.6: Distribución de grado de un grafo aleatorio del modelo de Erdős y Rényi. La red tiene  $N = 5000$  nodos y una probabilidad de enlace de  $p =$ .

Hay otro modelo, creado por Gilbert [31], de creación de redes aleatorias con  $n$  vértices y un número  $0 \leq m \leq [n(n-1)]/2$  de enlaces equiprobable. El método que se utiliza es la elección de un posible enlace y crearlo si no ha sido creado anteriormente, evitando bucles y enlaces múltiples, hasta completar el número de enlaces deseados.

- Grafo Regular. Es el grafo que contiene el mismo grado en todos sus vértices. Si  $k$  es el grado se le llama  $k$ -regular. Se puede ver un ejemplo en la figura 1.7.

La función de distribución,  $P(k)$ , de los grados en un grafo regular es un delta de Kronecker  $P(k') = \delta_{k',k} \equiv 1$  si  $k' = k$ , 0 si  $k' \neq k$ , donde  $k$  es el grado que comparten todos los vértices.

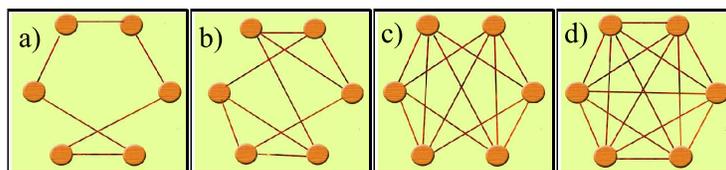


Figura 1.7: Grafos regulares a)2 – regular, b)3 – regular, c)4 – regular, d)5 – regular.

Para la creación de un grafo regular aleatorio de número par de  $N$  vértices y de grado  $k$  sin bucles ni aristas múltiples, se forman parejas aleatoriamente de nodos no enlazados del conjunto de vértices  $V$  del grafo  $G$ , hasta que todos los nodos tienen pareja y sin que un mismo nodo esté en más de una pareja, y se crean enlaces entre ellos, repitiéndose el proceso hasta que se llega al grado  $k$  deseado de los vértices.

Un ejemplo particular de grafo regular es el grafo completamente conexo o globalmente conexo. Este es una red en el que cada nodo se conecta con todos los demás excluyéndose a sí mismo.

- Árboles. Un árbol es un grafo conexo y acíclico que tiene las siguientes propiedades:
  - Entre cada par de vértices existe un camino único.
  - Toda arista es puente, eso quiere decir que si la arista es eliminada el grafo ya no es conexo.
  - Un árbol de  $n$  vértices tiene  $n - 1$  aristas.
  - Todo árbol no trivial tiene al menos dos hojas de grado 1.

Un árbol con raíz es un árbol en el cual se designa un vértice como raíz. Este tipo de grafos se representan de tal forma que la raíz esté por encima de los otros vértices. Los vértices subsiguientes se van colocando en niveles dependiendo de la distancia que los separe de la raíz. Se puede ver un ejemplo en la figura 1.8.

Se llama altura o profundidad de un árbol con raíz a la máxima distancia entre un vértice y la raíz.

Para definir algunos conceptos en la figura 1.8 se tiene que por ejemplo  $a$  es padre de  $b$  y  $c$ ;  $d, e, f$  y  $g$  son hijos de  $b$ ;  $k, e, b$  y  $a$  son ancestros de  $n$ ;  $n$  es descendiente de  $b$ ;  $d, j, n, f, l, o, p, i$  son hojas o vértices terminales. Si un vértice no es hoja ni raíz

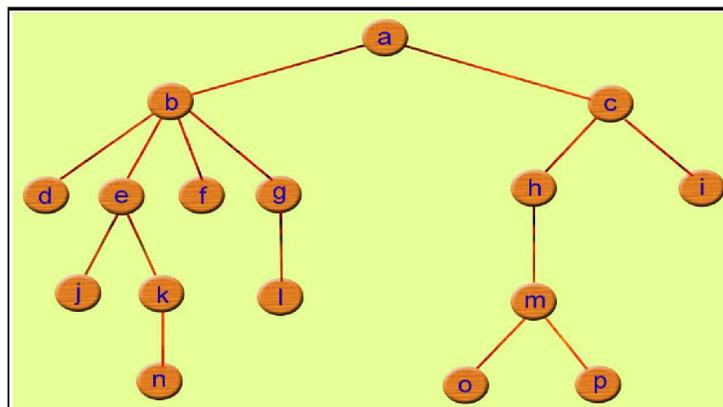


Figura 1.8: Árbol con raíz. El vértice  $a$  es la raíz; los nodos  $b$  y  $c$  forman el primer nivel;  $d$ ,  $e$ ,  $f$ ,  $g$ ,  $h$  e  $i$  el segundo;  $j$ ,  $k$ ,  $l$  y  $m$  el tercer nivel y  $n$ ,  $o$  y  $p$  el último nivel.

se le denomina vértice interno. El nivel de un vértice es la distancia que lo separa de la raíz.

Un árbol regular de ramificación  $R$  y profundidad  $L$  se forma con niveles de construcción,  $l = 0, 1, 2, \dots, L$ , con  $L$  la profundidad del árbol. El nivel  $l = 0$  se conforma por la raíz que tiene  $R$  enlaces conectando  $R$  hijos que se encuentran en el nivel de construcción 1. Cada uno de los hijos tienen  $R$  enlaces que conectan a  $R$  hijos nuevos, de forma tal que en cada nivel de construcción se producen  $R^l$  nodos. Los vértices del nivel de construcción  $l = L$  no tienen hijos, mientras que el vértice raíz en el nivel  $l = 0$  no tiene padre. Todos los demás vértices tienen un padre y  $R$  hijos. El número de nodos  $N$  de un árbol regular es:

$$N = (R^{L+1} - 1) / (R - 1). \quad (1.8)$$

En la figura (1.9) se puede ver un ejemplo de árbol regular.

La distribución,  $P(k)$ , de un árbol regular se divide en tres grupos. En el primero se encuentra la raíz en el nivel  $l = 0$  con  $k = R$ , el segundo grupo es el de los vértices en el nivel  $l = L$  con  $k = 1$  y por último los vértices restantes con  $k = R + 1$ . En la figura (1.10) se puedes ver un ejemplo de distribución de un árbol regular.

- Redes tipo mundo pequeño. Una red mundo pequeño es un grafo en el que la mayoría de los vértices no son vecinos entre sí pero en promedio la distancia entre ellos es

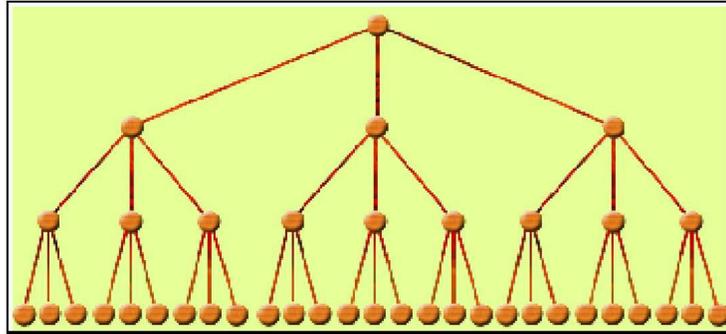


Figura 1.9: Árbol regular con  $R = 3$  y  $L = 3$ .

pueden pequeña. Las redes generadas aleatoriamente se caracterizan por tener una longitud característica y un coeficiente de agrupamiento pequeños. Las redes tipo mundo pequeño se parecen a las generadas aleatoriamente pero tienen un coeficiente de agrupamiento significativamente más alto. Al tener una longitud característica pequeña todos los vértices estarán conectados por caminos cortos y el coeficiente de agrupamiento grande genera que existan un número grande de subgrafos completamente conexos de grado tres.

Hay varias formas de construir redes tipo mundo pequeño. La utilizada en este trabajo fue ideada por Watts y Strogartz [32]. En el algoritmo, se empieza generando un anillo de  $N$  vértices donde cada nodo está conectado a sus  $\bar{k}$  vecinos más cercanos, con  $\bar{k}$  un número par. Se considera  $N \gg \bar{k} \gg \log(N) \gg 1$ . Luego con una probabilidad  $p$  cada enlace se redirige aleatoriamente a otro vértice, excluyendo bucles y aristas múltiples. Si  $p = 0$  se tiene como resultado una red regular con un gran número de ciclos pero con grandes distancias entre vértices. Si  $p \rightarrow 1$  la red se convierte en aleatoria con distancias cortas pero con pocos ciclos. Entre estos dos valores de  $p$  se forma el tipo de red mundo pequeño, como se puede ver en la figura 1.11. Una vez terminado el proceso la reconexión del grafo cambió su estructura pero el número de enlaces se mantiene constante a  $N\bar{k}/2$ .

En la figura 1.12 se puede ver el valor del agrupamiento Ec.(1.4) y el de la longitud característica Ec.(1.5) en el modelo de Watts y Strogartz para distintos valores de  $p$ . Para  $p = 0$ , es decir para una red regular, la longitud característica es larga con coeficiente de agrupamiento alto. En cambio para  $p = 1$ , la red es aleatoria y la

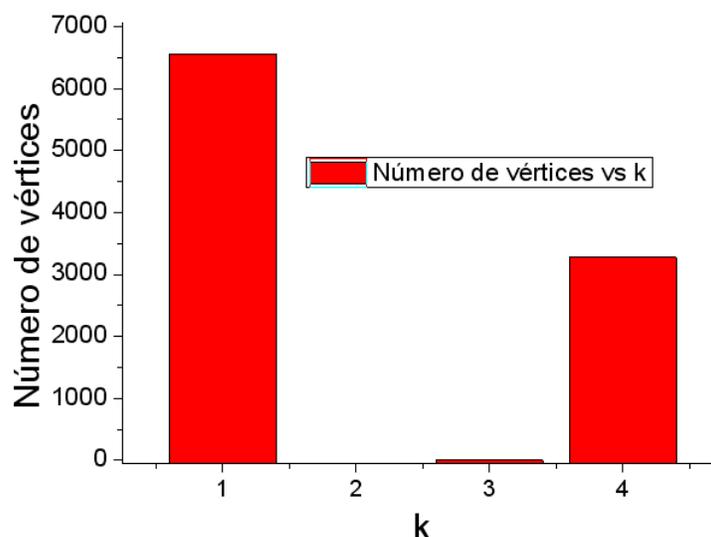


Figura 1.10: Distribución de frecuencia del número de enlaces por nodo  $f(P(k))$  de un árbol regular con  $N = 9841$ ,  $R = 3$  y  $L = 8$ . La primera columna es del grupo de  $l = L$  con  $k = 1$  y número de vértices= 6561, la segunda columna es la perteneciente al nivel  $l = 0$  con  $k = 3$  y número de vértices= 1, la última columna es de los niveles intermedios con  $k = 4$  y número de vértices= 3279.

longitud característica es corta con un coeficiente de agrupamiento bajo. Por último, con valores de  $p$  comprendidos entre  $10^{-3}$  y  $10^{-1}$  la red es tipo mundo pequeño y la longitud característica es corta con un coeficiente de agrupamiento alto.

La distribución de grado de la red pequeño mundo es parecida a una red aleatoria. El pico de la distribución de grados se encuentra en  $\bar{k}$  y tiene un decaimiento rápido a medida que nos alejamos de este valor. En la figura 1.13 se puede ver la distribución de grado de una red "mundo pequeño".

- Triángulo de Sierpinski. El término fractal, propuesto inicialmente por Benoît Mandelbrot en 1975 [17, 18], define a un objeto geométrico cuya estructura se repite a diferentes escalas. Los fractales se generan en procesos recursivos que producen estructuras autosimilares a cualquier escala de observación y de detalle infinito. El matemático polaco Waclaw Sierpinski introdujo un fractal, en 1919, a partir de un triángulo equilátero, que se toma como la iteración  $n = 0$ . En la primera iteración,  $n = 1$  se suprime el triángulo equilátero formado por los tres puntos medios del

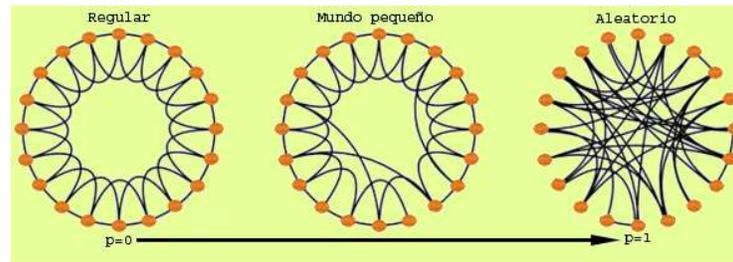


Figura 1.11: Modelo de Watts y Strogatz de formación de redes para distintos valores de  $p$  con  $N = 20$  y  $\bar{k} = 4$ .

triángulo inicial obteniéndose tres triángulos equiláteros iguales. En la siguiente iteración  $n = 2$  se repite el proceso suprimiendo los triángulos formados por los puntos medios de los triángulos anteriores, quedando así nueve triángulos equiláteros iguales. Se puede seguir este proceso tantas veces como se quiera. Se puede ver un ejemplo en la figura 1.14.

Si llevamos a  $n \rightarrow \infty$  el la figura formada se puede separar en  $3^n$  piezas autosimilares que aumentadas en un factor  $2^n$  nos devuelven la figura inicial. Esta autosimilitud a cualquier escala es un sello identificativo de un fractal. En tres dimensiones el triángulo de Sierpinski se construye con base en tetraedros. De la misma forma se pueden formar fractales con base en cuadrados, para dos dimensiones, y cubos, para tres dimensiones. Para formar una red utilizando fractales, en nuestro caso triángulos de Sierpinski, utilizamos las figuras creadas como nodos y se forman enlaces entre figuras que sean adyacentes. En figura 1.15, se puede ver una red de 27 nodos creada en una base de un triángulo de Sierpinski en dos dimensiones.

La distribución de grado de una red con base en el triángulo de Sierpinski generador es muy parecida a la de la red regular. En este caso para un triángulo de Sierpinski generalizado a dimensión  $d$  los vértices de los extremos exteriores conservan el número de enlaces  $k_e = d$  tras la primera interacción. Todos los demás vértices tendrán un mismo  $k_s = d + 1$ .

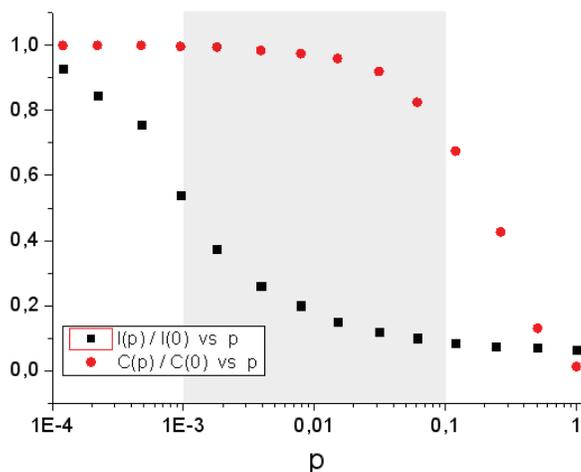


Figura 1.12: Agrupamiento y longitud característica en el modelo de Watts y Strogatz para distintos valores de  $p$ . La zona sombreada indica aproximadamente el rango de lo que se considera red tipo mundo pequeño.

## 1.2. Algoritmo genético

Durante las últimas décadas se ha estado intentando buscar herramientas para resolver problemas que no se pueden enfocar desde la vista clásica. Los métodos clásicos suelen ser elegantes, eficientes y precisos, pero con frecuencia presentan limitaciones en su aplicación práctica. Para atacar algunos problemas que no pueden ser abordados de forma clásica se han desarrollado los métodos evolutivos que se basan en la teoría de Darwin y en conceptos básicos de la biología genética y son una herramienta en auge para trabajar con casos de la vida real.

Los algoritmos genéticos forman parte de la computación evolutiva [33, 34, 35, 36, 37]. Estos son apropiados para resolver problemas donde el dominio de la solución es demasiado extenso. Los algoritmos genéticos tienen como finalidad buscar un conjunto de soluciones basadas en un nivel de aptitud (fitness).

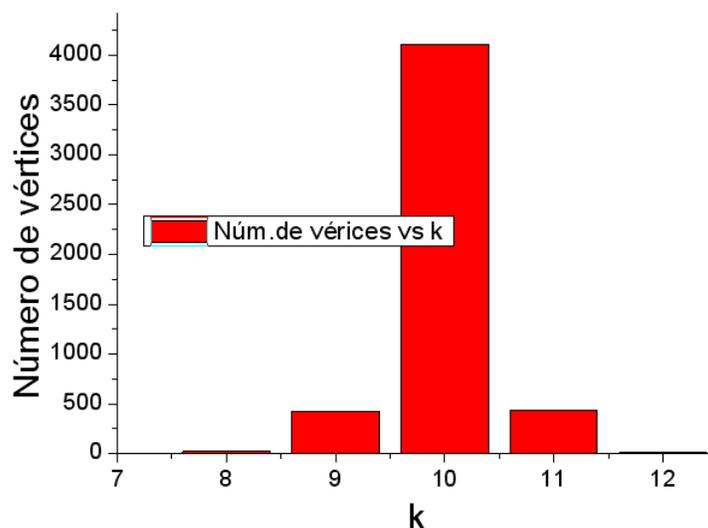


Figura 1.13: Distribución de grado en una red tipo mundo pequeño. La red es de  $N = 5000$  nodos,  $n = 10$  y  $p =$ .

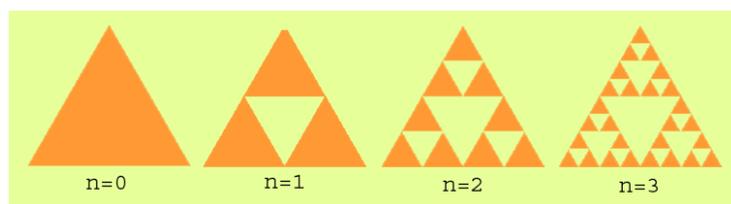


Figura 1.14: Tres iteraciones del triángulo de Sierpinski.

### 1.2.1. El lenguaje de los algoritmos genéticos

Los algoritmos genéticos tienen su origen en la genética natural y la ciencia computacional por lo que en los algoritmos genéticos hay una mezcla de los dos lenguajes.

El algoritmo genético opera iterativamente actualizando al conjunto de hipótesis que llamamos población. En cada iteración toda la población es evaluada de acuerdo a una función objetivo. Se genera una nueva población de acuerdo al valor alcanzado por la función objetivo. Algunos individuos pasan intactos y otros son generados por medio de la reproducción de los individuos seleccionados. Cuando se llega a la condición de parada se estudian los individuos mejores para analizar los resultados. En la tabla 1.2 se puede ver el prototipo de un algoritmo genético simple:

Los cromosomas o secuencias se combinan para formar la prescripción genética total

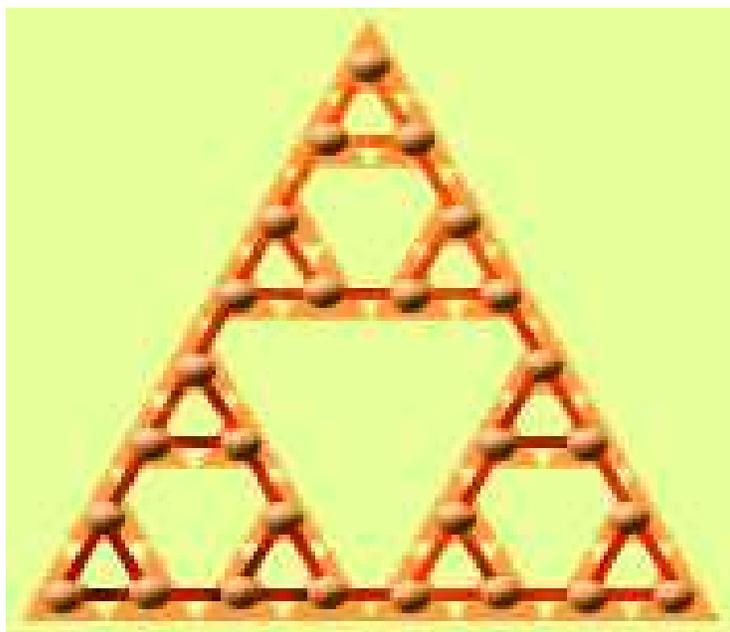


Figura 1.15: Red formada utilizando como base el triángulo de Sierpinski.

de los nuevos individuos. El paquete genético completo se llama genotipo o estructura. Los cromosomas están formados por genes o caracteres que pueden tomar ciertos valores según el problema con el que se esté trabajando. En la figura 1.16 se puede ver un ejemplo de un genotipo con sus componentes.

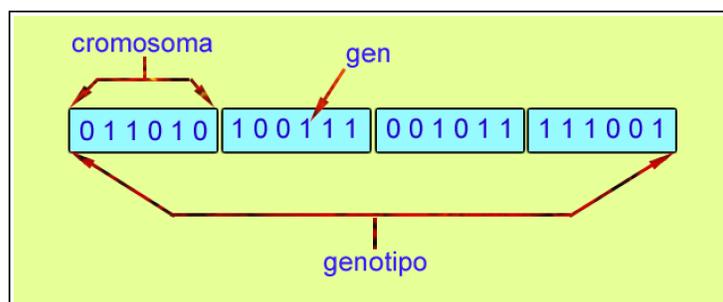


Figura 1.16: Genotipo con sus componentes.

Inicialización del contador de iteraciones.  
 Generación de la población inicial.  
 Para cada individuo:  
 - Evaluar con la función objetivo.  
 - Actualizar el criterio de finalización.  
 Mientras (no se cumpla el criterio de finalización) hacer:  
 - Incrementar el contador de iteraciones.  
 - Seleccionar los nuevos padres.  
 - Recombinar los padres.  
 - Mutar los individuos recombinados.  
 - Seleccionar la nueva población de individuos.

tabla 1.2: Esquema del algoritmo genético.

### 1.2.2. Tipos de representación

Durante los primeros años, el tipo de representación que se utilizaba era el alfabeto binario ( $\Omega = 0, 1$ ). Esta representación se adaptaba perfectamente al tipo de operaciones y operadores que se utilizaban en ese tiempo para un algoritmo genético. Sin embargo, el alfabeto binario no siempre es el apropiado para algunas situaciones. Una representación debe ser capaz de abarcar las características constituyentes de un conjunto de soluciones. Básicamente hay tres tipos de representación:

- Representación binaria,  $\Omega = 0, 1$ . El gen puede tomar valores de 0 ó 1.

1 0 0 1 1 0 1

- Representación entera,  $\Omega \in Z$ . El gen puede tomar valores enteros.

4 5 -9 0 1 4

- Representación real,  $\Omega \in R$ . El gen puede tomar valores reales.

0,59 -5,36 0 1 9,2 4,75

### 1.2.3. Población

Todo algoritmo genético se inicia generando una población inicial de  $n$  individuos. Si una población es pequeña se corre el riesgo de no cubrir por completo el espacio de búsqueda, mientras que una población excesivamente grande puede generar problemas de costo computacional. Cada individuo está formado por cadenas de caracteres (cromosomas) que pueden ser los elementos de cualquier alfabeto ( $\Omega$ ).

$$\text{cromosoma} = x_1, x_2, \dots, x_n \quad x_i \in \Omega \quad 1 \leq i \leq n \quad (1.9)$$

donde, a cada  $a_i$  se le llama gen. Para no perder exactitud en la solución, en el momento de optimizar una función, hay que asegurarse que la longitud del cromosoma permite codificar todos los puntos dentro del intervalo de la función. Por ejemplo, en un alfabeto binario ( $\Omega = 0, 1$ ), si tenemos cromosomas de longitud  $n$  podemos codificar  $2^n$  valores.

Normalmente, la formación de los individuos de la población inicial se efectúa de forma aleatoria donde cada gen tendría uno de los posibles valores del alfabeto con una cierta probabilidad. Si la población inicial se eligiera con cierta técnica de optimización podría resultar en una convergencia prematura hacia un óptimo local en vez del buscado.

### 1.2.4. Función objetivo

Un aspecto que resulta crucial en la creación del algoritmo genético es la determinación adecuada de la función objetivo. Para dos individuos que se encuentren cercanos en el espacio de búsqueda de la función objetivo, sus respectivos valores deberían ser similares. Otra dificultad podría ser la existencia de muchos óptimos locales o que el óptimo se encuentre aislado.

Si el algoritmo genético es correctamente implementado, la población evolucionará a lo largo de las generaciones de forma que la aptitud media de la población se irá incrementando hacia el óptimo global. El concepto de convergencia está relacionado con

la progresión hacia la uniformidad en la aptitud global. Se dice que un cromosoma ha convergido cuando al menos el 95 % de la población comparte el mismo valor para dicho cromosoma, y se dice que la población converge cuando todos los cromosomas han convergido. Se puede generalizar al caso en que sólo una parte de la población haya convergido. En la figura 1.17 se puede ver que a medida que pasan las generaciones la aptitud media se va aproximando a la aptitud del mejor individuo. Los algoritmos genéticos pueden presentar dos problemas de convergencia, el primero de convergencia prematura surge frecuentemente cuando el rango de la función objetivo es proporcional a la elección de los individuos más aptos. En ese caso suelen existir individuos con un valor de evaluación muy superior al resto y que van dominando la población a medida que avanza el algoritmo. Una solución es la transformación de la función objetivo como la de disminuir el rango de la función objetivo de forma que dichos individuos no lleguen a dominar la población. En el problema de la convergencia lenta sucede lo contrario; la solución, de manera análoga a la anterior, se logra efectuando una expansión del rango de la función objetivo.

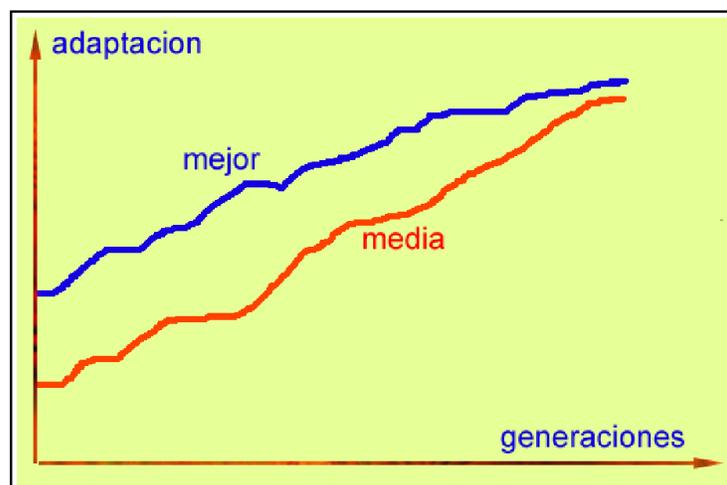


Figura 1.17: Adaptación a la función objetivo del mejor individuo y de la media de la población.

Goldberg y Richardson [38] proponen un método donde se modifica la función objetivo de cada individuo de acuerdo a la similitud que tengan entre sí; de esta forma se gana diversidad en la población. Cuanto mayor sea la similitud entre los individuos menor será la probabilidad de que sean escogidos como padres aumentando así la

probabilidad de elección de individuos más aislados.

### 1.2.5. Operadores básicos de un algoritmo genético

Los algoritmos genéticos constan fundamentalmente de los siguientes operadores: reproducción, recombinación, mutación y traslocación.

- Operador reproducción. Es el encargado de conservar aquellas características de los individuos que se consideren valiosas a lo largo de las generaciones. Dependiendo del arreglo de los genes, cada individuo tendrá un valor dado por la función evaluadora. La función evaluadora marca el acercamiento del individuo a la solución que se quiere alcanzar, marcando la aptitud para la supervivencia de cada individuo en la próxima generación.

Después de evaluar la población, hay que seleccionar aquellos individuos que se reproducirán. Veremos a continuación algunas opciones:

- Aleatoriamente. Se eligen al azar los individuos que se reproducirán sin importar el valor dado por la función evaluadora.
- Los mejores. Los  $m$  individuos que tengan el valor más alto se reproducen, donde  $m$  es parámetro del algoritmo. Esta opción puede venir acompañada por un umbral en el valor requerido. Solo los que pasen de este umbral se reproducirán.
- Selección proporcional a la aptitud. La probabilidad de reproducirse es proporcional al valor obtenido por la función evaluadora. El comportamiento es similar al de una ruleta sólo que los que tienen un valor de evaluación más alto tienen más probabilidad de ser escogidos.
- Selección proporcional al rango. Los individuos son evaluados por la función objetivo para después ser ordenados de mayor a menor según el valor obtenido. El lugar que ocupa el individuo en esta escala se le denomina rango. El mejor individuo tendrá valor 1 y el peor tendrá el valor  $\lambda$ . En la figura 1.18 se ve que la repartición de probabilidad utilizando el rango es más uniforme que en la de probabilidad por aptitud. Tenemos que:

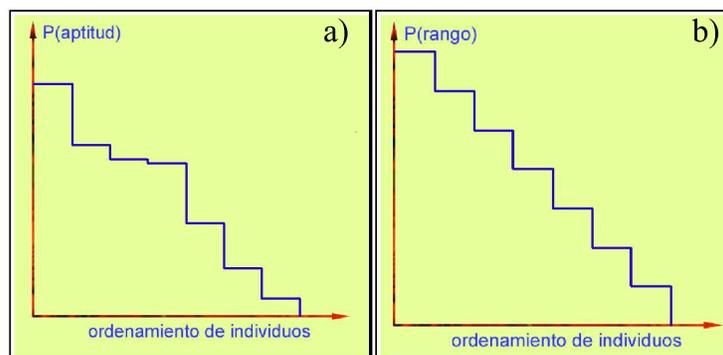


Figura 1.18: Selección de padres proporcional a) a la aptitud y b) al rango de la aptitud.

- Los  $m$  mejores y los  $n$  peores. Los  $m$  individuos con valor más alto de evaluación se emparejan con los  $n$  individuos con menor valor.
  - Todos. Todos los individuos se emparejan al azar para la reproducción.
  - Sólo el mejor. El individuo que obtenga el mayor valor es el único que se reproduce.
- Operador recombinación. A los individuos que se han elegido para la reproducción se les aplica el operador de recombinación. En primer lugar, se emparejan los individuos al azar. Seguidamente, cada par de individuos se recombina. Las técnicas básicas de recombinación son:
    - Recombinación básica. Se elige un número al azar  $k \in 1, \dots, n$ , con  $n$  el tamaño del cromosoma, y se hace un intercambio de contenido entre los dos individuos desde el gen  $k$  hasta el final como se puede ver en la figura 1.19.

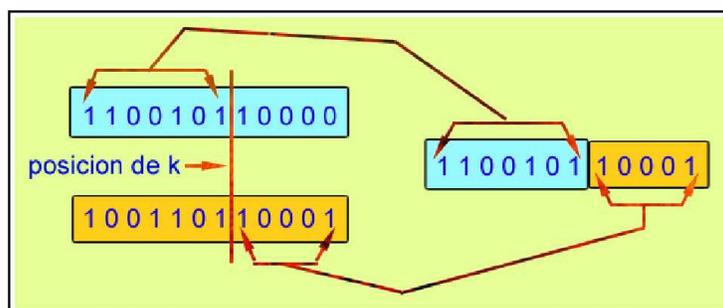


Figura 1.19: Recombinación básica de los padres (amarillo y azul) a partir de una posición  $k$ .

- Recombinación multipunto. Es igual al anterior sólo que hay varios puntos de

intercambio. Podemos ver un ejemplo en la figura 1.20.

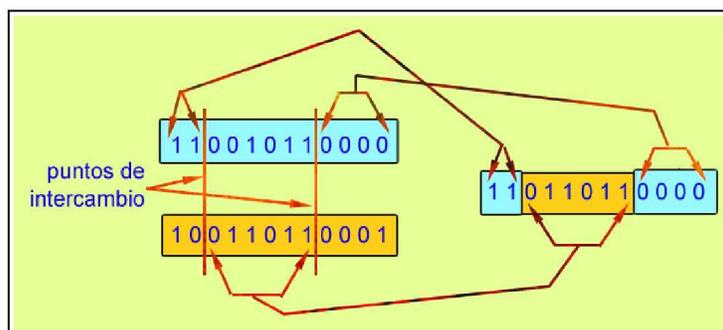


Figura 1.20: Recombinación multipunto de dos padres (amarillo y azul) con dos posiciones de intercambio.

- Recombinación uniforme. Se eligen de forma aleatoria las posiciones de los genes de los dos individuos (padres) para la formación de los hijos. En la figura 1.21 podemos ver que se elige de forma aleatoria la posición del gen en los padres y ésta se conserva en el hijo.

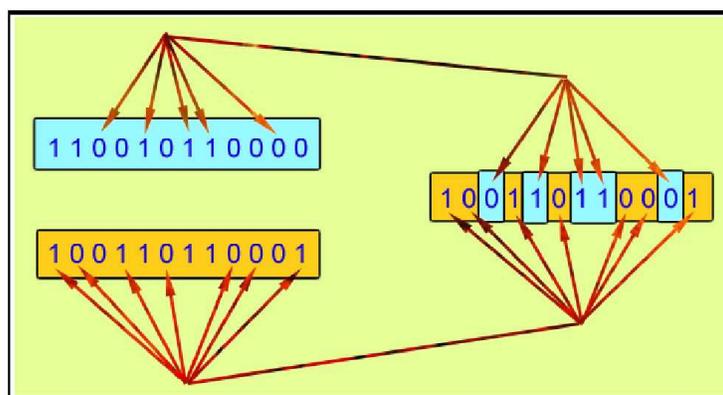


Figura 1.21: Recombinación uniforme de dos padres (amarillo y azul).

- Recombinación real. Ésta es sólo aplicable a individuos con genes reales,  $Q$ . En ella se elige un número  $r \in [0, 1]$  que combina a los padres de la siguiente forma:

$$Q_{hijo} = rQ_{padre1} + (1 - r)Q_{padre2} . \quad (1.10)$$

- Operador mutación. Este operador proporciona cierta aleatoriedad a los individuos de la nueva generación. El objetivo del operador mutación es el de modificar a cierto número de genes de unos pocos individuos con el fin de aumentar la diversidad dentro de la población. Algunas razones por las cuales se debe implementar la mutación son las siguientes:
  - Desbloqueo del algoritmo. Si el algoritmo se queda bloqueado dentro de un máximo o un mínimo local, la mutación puede desbloquearlo al incorporar más diversidad en otras zonas del espacio.
  - Poblaciones degeneradas. Puede que en algún momento surja un individuo muy bueno que, a lo largo de las generaciones, domine la población.
  - Saltos evolutivos. La aparición de un fenotipo especialmente valioso es muy poco probable dentro de un algoritmo genético sin mutaciones.

Existen varias técnicas de mutación según mostraremos a continuación:

- Mutación de gen. Para cada individuo hay una probabilidad de que se produzca la mutación de un solo gen. Al producirse la mutación el valor del gen se invierte como se ve en la figura 1.22.

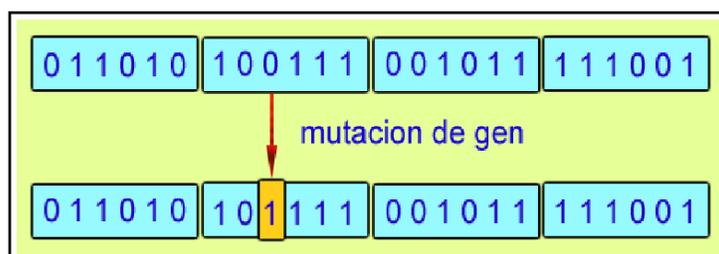


Figura 1.22: Mutación de gen.

- Mutación multigen. Para los individuos de la población todos los genes tienen probabilidad de mutar.
- Mutación de cromosoma. Es igual a la mutación de gen pero cambia un cromosoma completo. En el cromosoma se puede cambiar desde un gen hasta el cromosoma completo. En la figura 1.23 se ve un ejemplo.
- Mutación multicromosoma. El procedimiento es el mismo que en la mutación de cromosoma solo que cambian varios cromosomas.

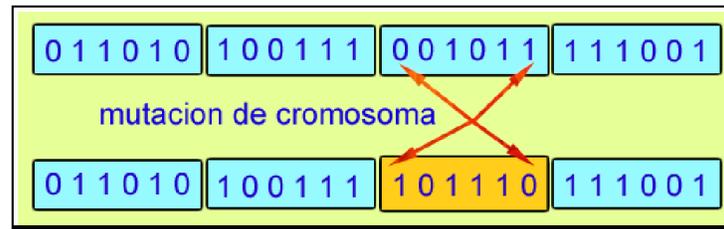


Figura 1.23: Mutación de cromosoma.

- Mutación por traslocación. En esta mutación se toman dos genes o cromosomas aleatoriamente y se intercambian. Esto es útil cuando se desean conservar ciertas propiedades que se perderían con otro tipo de operador. Se puede ver un ejemplo en la figura (1.24).

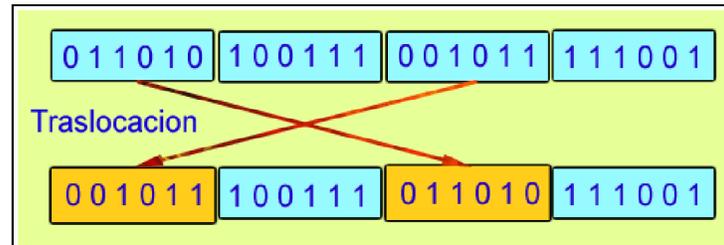


Figura 1.24: Mutación por traslocación de cromosomas.

- Mutación por barajado. Se toman dos cromosomas y se barajan de forma aleatoria los genes comprendidos entre los dos.
- Reemplazo de la población y condición de parada. Tras la acción de los operadores de reproducción, recombinación y mutación, se generan cierto número de nuevos individuos que entrarán en la próxima generación de la población. El reemplazo de la población en la nueva generación se puede hacer de diversas formas:
  - Tras la creación de cierta cantidad, previamente concebida, de nuevos individuos, se elimina la misma cantidad de los individuos de la población anterior con peor valor de evaluación.
  - Con la creación de cada nuevo individuo se elimina al peor adaptado de la nueva población.

- Se crean suficientes nuevos individuos para que ellos, por sí solos, formen la nueva generación.

La condición de parada puede venir determinada por varios criterios; a continuación les mostraremos algunos:

- Un número máximo de generaciones.
- Un tiempo máximo de ejecución del programa.
- Pérdida de la diversidad.
- Estado de evolución de la población.
- Ausencia de mejoras tras un cierto número de interacciones.

Generalmente se suele usar una combinación de estos criterios. Se usa un algoritmo con límite para las generaciones con una condición de parada si no hay mejoras tras cierto número de interacciones.

### 1.2.6. Parámetros de un algoritmo genético

- Porcentaje de recombinación. Indica con qué frecuencia se recombinarán los individuos. Si el porcentaje es de 0% los hijos serán idénticos a los padres y sólo se alterarán por mutación. Si es de 100% todos los hijos serán creados mediante recombinación de los individuos de la generación previa. Normalmente es recomendable que algunos individuos pasen intactos a la siguiente generación.
- Porcentaje de mutación. Este porcentaje establece cuál de los individuos será mutado. Si el porcentaje es de 0% los individuos generados después de la recombinación no serán mutados. Si es de 100% todos los individuos de la nueva generación serán mutados. El objetivo de la mutación es impedir que el algoritmo caiga en un máximo o mínimo local. Es recomendable que la mutación exista pero tenga baja probabilidad ya que si tiene una probabilidad muy alta pasa a ser una búsqueda aleatoria.
- Tamaño de población. Establece cuántos individuos habrá en cada generación. Si el tamaño es muy bajo el algoritmo tiene pocas probabilidades de evolucionar

y hay un punto en el que por mucho que se aumente la población los resultados no mejoran.

- Número de generaciones. Debe ser lo suficientemente grande para que el algoritmo pueda converger hacia el óptimo global pero cuidando de que no haya desperdicio del costo computacional. Este parámetro debe estar condicionado a la convergencia del algoritmo de forma que, en principio, es difícil de determinar.
- Tamaño del individuo. Debe cubrir todos los puntos dentro del espacio de búsqueda. Un tamaño insuficiente podría acarrear el no alcanzar el óptimo global de la función objetivo.

### 1.3. Mapas unidimensionales

Los mapas, que se conocen también como ecuaciones diferenciales finitas, relaciones de recursión, mapas iterados o mapas dinámicos; son relaciones unidimensionales de sistemas dinámicos en los que la variable tiempo es discreta y el estado es continuo. Los mapas sirven como modelos en áreas como electrónica digital, economía, caos y también se usan como representaciones discretas de ecuaciones diferenciales [39, 40].

El mapa unidimensional general es de la siguiente forma:

$$x_{t+1} = f(x_t, r), \quad (1.11)$$

donde  $x_t$  es el estado del mapa en un tiempo discreto  $t$ ,  $r$  es una constante que determina el grado de no-linealidad del mapa llamada parámetro de control y  $f(x_t, r)$  es una función de mapeo que depende de  $x_t$  y de  $r$ . La secuencia  $x_0, x_1, x_2, \dots$  se llama trayectoria u órbita correspondiente a la condición inicial  $x_0$  [41].

#### 1.3.1. Puntos fijos y estabilidad local

Se dice que  $x^*$  es un punto fijo del mapa si

$$f(x^*) = x^* . \quad (1.12)$$

Dado  $x_i = x^*$ , tenemos que  $x_{t+1} = f(x_t) = f(x^*) = x^*$  para todo  $i \geq t$ , por lo tanto si la órbita cae en un punto fijo seguirá en el durante sucesivas iteraciones.

La estabilidad de un punto fijo  $x^*$  se estudia considerando una órbita próxima  $x_t = x^* + \eta_t$  y determinamos si  $\eta_t$  decrece (es atraída) o crece (es repelida) al aumentar  $t$ . De la definición de mapa se tiene que:

$$x_{t+1} = x^* + \eta_{t+1} = f(x_t) = f(x^* + \eta_t) = f(x^*) + f'(x^*)\eta_t + O(\eta_t^2), \quad (1.13)$$

donde  $O(\eta_t^2)$  es un término residual del orden de  $\eta_t^2$ . Usando  $f(x^*) = x^*$ , tenemos:

$$\eta_{t+1} = f'(x^*)\eta_t + O(\eta_t^2), \quad (1.14)$$

si despreciamos  $O(\eta_t^2)$  tendríamos un mapa linealizado  $\eta_{t+1} = f'(x^*)\eta_t$ . Se define como valor propio o multiplicador a la pendiente en el punto fijo:

$$\rho \equiv f'(x^*). \quad (1.15)$$

La estabilidad del punto fijo  $x^*$  depende del valor de  $\rho$ . Los miembros del mapa linealizado son:

$$\eta_1 = \rho\eta_0, \quad \eta_2 = \rho\eta_1 = \rho^2\eta_0, \quad \dots, \quad \eta_t = \rho^t\eta_0, \quad (1.16)$$

por lo tanto:

- Si  $|\rho| < 1$ ,  $\eta_t \rightarrow 0$  para  $t \rightarrow \infty$ , el punto fijo es linealmente estable y se le dice que es atractor,  $x_e^*$ .
- Si  $\rho = 0$  el punto fijo es superestable.
- Si  $|\rho| > 1$  el punto fijo es inestable,  $x_i^*$ .
- Si  $|\rho| = 1$  el punto fijo es crítico,  $x_c^*$ . El parámetro de control,  $r$ , en este punto se le denomina  $r_c$ .

Si el parámetro de control  $r$  varía, generalmente, también cambia el valor propio del punto fijo llegando para ciertos valores de  $r$  a perderse la condición de estabilidad  $|\rho| < 1$ . Para este valor del parámetro de control la trayectoria deja de tender hacia

el atractor  $x^*$  y se dice que la “órbita período 1” se hizo inestable. En algunos casos, como en el mapa logístico  $f(x) = rx(1-x)$ , en este punto emerge una “órbita período 2” que corresponde a los dos puntos estables:

$$x_1^* = f(x_0^*, r) \quad , \quad x_0^* = f(x_1^*, r) \quad (1.17)$$

A este fenómeno se le llama duplicación o dobladura del período. En el mapa logístico a medida que sigue creciendo  $r$  se repite el fenómeno de bifurcación hasta que no existe ningún punto atractor y coexisten infinitos puntos repulsores con cualquier período. A la dinámica producida por esta condición se le denomina caos. En otros mapas, como en el mapa logarítmico

$$x_{t+1} = f(x_t) = r + \ln|x_t|, \quad (1.18)$$

surge la transición al caos sin dobladuras del período. Se puede ver en la figura 1.25 el diagrama de bifurcación del mapa logarítmico. El punto fijo  $x_{1e}^*$  pasa a ser inestable en  $r_{1c} = -1$  volviendo la estabilidad en  $r_{2c} = 1$  formándose el punto fijo  $x_{2e}^*$ , por lo tanto el caos ocurre en el intervalo del parámetro  $r \in [-1, 1]$ .

### 1.3.2. Órbitas

Una órbita en un mapa unidimensional es el conjunto de puntos resultantes de iteraciones  $\{x, f(x), f^2(x), \dots\}$  donde el valor inicial,  $x_0$ , de una órbita es el punto donde inicia la órbita. Una grafica útil para ver el comportamiento de las órbitas de un mapa unidimensional es la gráfica de la telaraña. Los pasos para realizar esta gráfica son:

1. Se grafican el mapa  $f(x)$  y el bisector  $y = x$ .
2. Se traza una línea vertical desde el punto inicial,  $x_0$ , en el eje  $x$  hasta la curva  $f(x)$ .
3. Desde el punto de cruce en la curva  $f(x)$  se traza una línea horizontal hasta  $y = x$ .
4. Se traza una vertical desde el punto donde se tocó a  $y = x$  hasta  $f(x)$ .

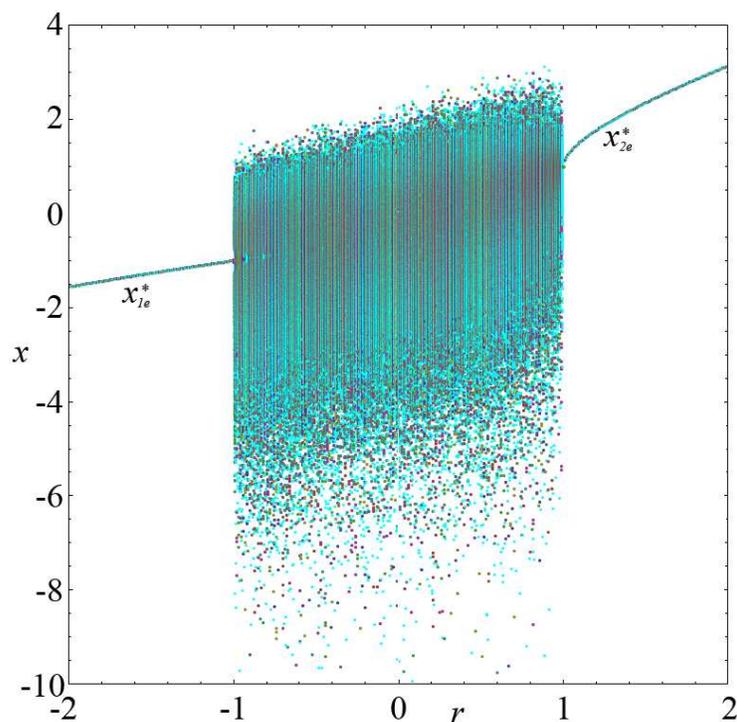


Figura 1.25: Diagrama de bifurcación de la iteración del mapa logarítmico como función de  $r$ . Con cada incremento de 0,005 en el valor de  $r$  se descartan 10000 iteraciones y se grafican 1000.

5. Se repiten los pasos 3 y 4 tantas veces como sean necesarios.

En la figura 1.26 se puede ver un ejemplo de una órbita del mapa logarítmico.

### 1.3.3. Intermitencia

Pomeau y Manneville, en 1979, resolvieron numéricamente las ecuaciones diferenciales del modelo de Lorentz encontrando el fenómeno de la intermitencia [42, 43, 44]. La intermitencia es la ocurrencia de una señal que alterna entre fases regulares largas y ráfagas irregulares cortas. El número de ráfagas caóticas se incrementan al variar un parámetro externo, por lo que la intermitencia ofrece una ruta continua desde el comportamiento regular al caótico. Hay tres tipos de intermitencia descubiertas por Pomeau y Manneville de las cuales hablaremos del tipo I y del tipo III, que son las que existen para el mapa logarítmico. Para mayor información de la intermitencia

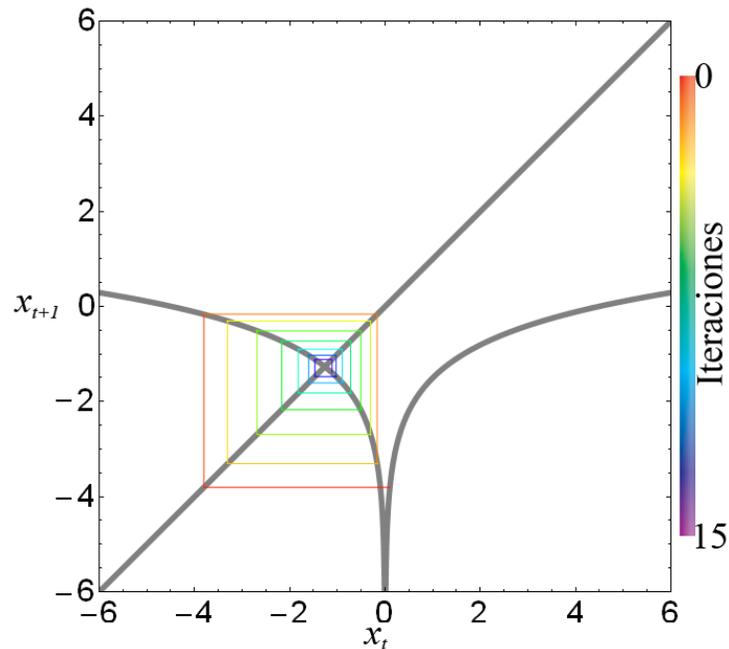


Figura 1.26: Órbita del mapa logarítmico con  $r = -1,5$  y  $x_0 = 0,1$  donde se muestran 15 iteraciones.

tipo II se puede consultar [45].

La intermitencia tipo I está caracterizada por una bifurcación tangente inversa con dos puntos fijos, uno estable y otro inestable. En la figura 1.27 se pueden ver las órbitas periódicas del mapa logarítmico que cumplen con  $f(x) = x$ . Los dos puntos fijos  $x_{2e}^*$  y  $x_{2i}^*$  dan paso al caos a partir de  $x_{2c}^*$ . Para  $r_{1c} < r < r_{2c}$  el mapa no tiene puntos fijos estables. Sin embargo un punto fijo inestable se muestra mientras el movimiento de la trayectoria se encuentra en la vecindad de  $x_{2c}$  como se ve en la figura 1.28. Para  $r \approx r_{2c}$  pero menor, un número de iteraciones son necesarias para moverse a través del canal estrecho entre el mapa y el bisector. Esto conduce a una región regular que se torna caótica cuando la trayectoria sale del canal hasta que se produce la reinyección en la vecindad de  $x_{2c}^*$  empezando una nueva fase regular.

En la figura 1.29 se puede ver el comportamiento de la órbita en el mapa logarítmico para distintos valores de  $r$ . Para  $r_{2c} = 1$  vemos que la órbita llega rápidamente al punto fijo manteniéndose en él indefinidamente. Para  $r$  ligeramente inferior a  $r_{2c}$  se produce el fenómeno de intermitencia debido a los pasos por el canal entre el mapa

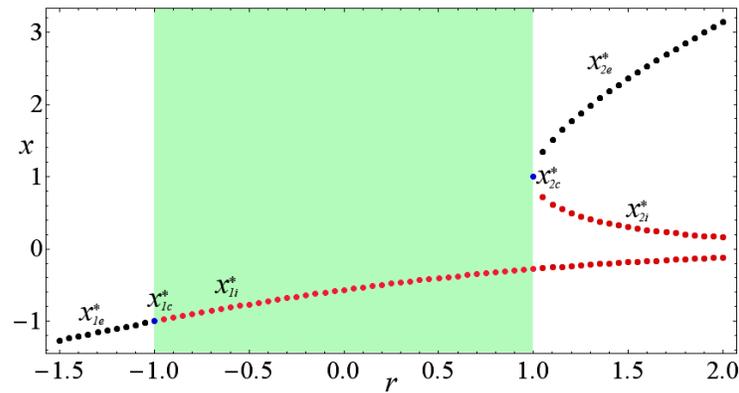


Figura 1.27: Puntos fijos del mapa logarítmico que cumplen con  $f(x) = x$ . Los puntos negros indican puntos fijos estables y los rojos puntos fijos inestables. Se grafica en intervalos de  $r$  de 0,05. La zona verde indica comportamiento caótico.

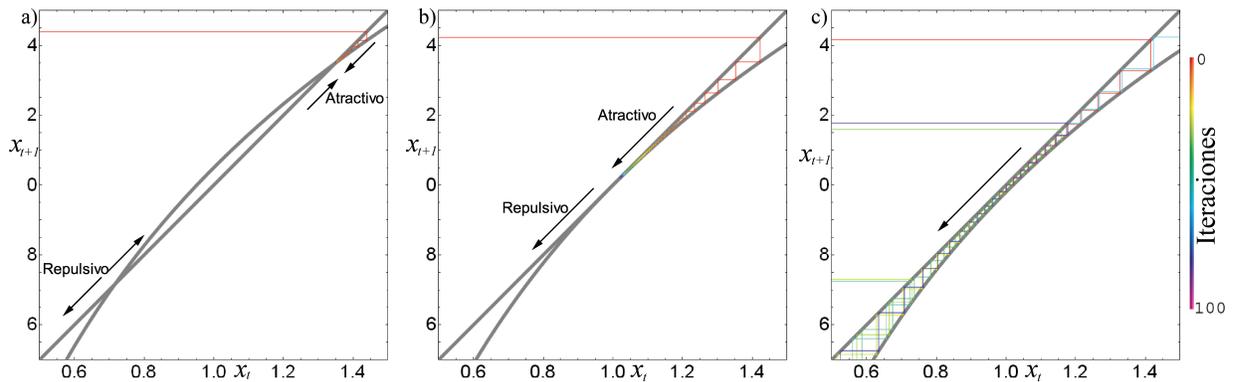


Figura 1.28: Acercamiento de la órbita del mapa logarítmico en la región donde se produce la intermitencia tipo I para distintos valores de  $r$ . En a) se tiene  $r = 1,05$  con  $\lambda < 0$ , en b)  $r = 1$  con  $\lambda = 0$  y en c)  $r = 0,98$  con  $\lambda > 0$ . En las tres figuras se graficaron 100 iteraciones.

y el bisector. La intermitencia se produce a través de un incremento o decrecimiento monótono hasta el estallido caótico como podemos ver en la ampliación. Para  $r < r_{2c}$  el canal es más amplio por lo que la órbita pasa rápidamente a su través predominando el comportamiento caótico.

La intermitencia tipo III aparece simultáneamente con una bifurcación de duplicación del periodo. En la figura 1.30 se puede ver, para el mapa logarítmico, que el punto fijo  $x_{1e}^*$  es estable para  $r < -1$  y con órbita periodo 1. En  $x_{1c}^*$ , con  $r = -1$  se inicia el fenómeno de intermitencia junto con una duplicación del periodo que va

umentando de amplitud hasta que rompe en el caos por un breve lapso. Esto se debe a que el punto fijo  $x_{1i}^*$  es mínimamente repulsivo en  $r = 1$ , como se ve en la figura 1.31. Cuando la órbita pasa muy cerca de  $x_{1i}^*$  tarda tiempo en alejarse a lo que se debe el crecimiento en la amplitud. Una vez alejada la órbita de  $x_{1i}^*$  se comporta de forma caótica hasta que vuelve a caer en sus cercanías. A medida que  $r$  crece el punto fijo  $x_{1i}^*$  se torna más y más repulsivo por lo que la órbita pasa poco tiempo en sus cercanías predominando el comportamiento caótico.

### 1.3.4. Sensibilidad a las condiciones iniciales y exponente de Liapunov

Dos órbitas con puntos iniciales cercanos que se encuentren dentro de un intervalo periódico suelen mantenerse aproximadamente a la misma distancia. Si se encuentran en una zona caótica no hay puntos atractores, por lo tanto las órbitas de los dos puntos iniciales tienen un futuro totalmente diferente. A esto se le llama sensibilidad a las condiciones iniciales y está relacionado al valor absoluto de la derivada del mapa como veremos a continuación.

Haciendo la expansión de Taylor de una función suave cualquiera  $f$ , tenemos que:

$$f(x + h) = f(x) + f'(x)h + O(h) , \quad (1.19)$$

que arreglando un poco la ecuación nos queda:

$$f(x + h) - f(x) = f'(x)h + O(h). \quad (1.20)$$

Esto quiere decir que si dos puntos están en un principio a una distancia  $|h|$  en la siguiente iteración estarán a una distancia  $|f'(x)h|$  si se desprecia  $O(h)$ , esto es, si  $|f'(x)| > 1$  entonces la distancia que los separe será mayor que  $|h|$ . Por lo tanto el que un punto  $x + h$  se acerque o se aleje de  $x$  depende del valor de la derivada  $f'(x)$ .

Sea una condición inicial  $x_0$  y otra extremadamente próxima  $x_0 + \delta_0$ . Supongamos que la separación después de  $n$  iteraciones es  $|\delta_n| = |\delta_0| e^{n\lambda}$ . Despejando  $\lambda$  y tomando en cuenta que  $\delta_n = f^n(x_0 + \delta_0) - f^n(x_0)$ , se tiene que:

$$\lambda = \frac{1}{n} \ln \left| \frac{\delta_n}{\delta_0} \right| = \frac{1}{n} \ln \left| \frac{f^n(x_0 + \delta_0) - f^n(x_0)}{\delta_0} \right| = \frac{1}{n} \ln |(f^n)'(x_0)| , \quad (1.21)$$

de donde se supone el límite  $\delta_0 = 0$ . Si esta expresión tiene límite para  $n \rightarrow \infty$ , entonces a este límite,

$$\lambda = \lim_{n \rightarrow \infty} \frac{1}{n} \ln |(f^n)'(x_0)|, \quad (1.22)$$

se le llama exponente de Liapunov para la órbita que comienza en  $x_0$ .

La definición dada depende de  $x_0$  pero se espera el mismo resultado para cualquier órbita en la cuenca de atracción de un atractor y se tiene que  $\lambda > 0$  para los atractores caóticos y  $\lambda < 0$  para puntos fijos y ciclos estables.

La figura 1.32 muestra el exponente de Liapunov  $\lambda$  en el mapa logarítmico. Es positivo para todo el intervalo de  $r \in [-1, 1]$ .

## 1.4. Redes de mapas acoplados

Las redes de mapas acoplados (CML, coupled map lattice en inglés) se introdujeron simultánea e independientemente, en los ochenta, por K.Kaneko, R.Kapral y S.Kuznetsov [4, 5, 6, 7, 8, 9]. Una red de mapas acoplados es un sistema dinámico con espacio discreto, tiempo discreto y espacio de estado continuo. Se puede pensar en las redes de mapas acoplados como generalizaciones de los autómatas celulares (CA) cuyo espacio de estado es discreto.

La dinámica de las redes de mapas acoplados consiste en dos estados independientes: las dinámicas locales y las dinámicas acopladoras. Para ello se aplica un mapa local unidimensional,  $f(x)$ , en cada nodo de la red para luego acoplar las dinámicas por medio de una suma de los pesos sobre los  $\nu_i$  vecinos del elemento  $i$ . En nuestro trabajo se usa una red de mapas acoplados en los que un nodo interactúa con cualquier otro nodo con el que esté enlazado directamente. Esto se expresa de la siguiente manera:

$$x_{t+1}(i) = (1 - \epsilon)f(x_t(i)) + \frac{\epsilon}{N_i} \sum_{j \in \nu_i} f(x_t(j)), \quad (1.23)$$

donde,  $x_t(i)$  es el estado del elemento  $i$  en el tiempo discreto  $t$ ,  $i$  indexa los nodos con  $i = 1, 2, \dots, N$ ,  $\epsilon$  representa el coeficiente de difusión,  $f(x)$  es un mapa no lineal caótico y representa la dinámica local de cada nodo,  $\nu_i$  es el conjunto de nodos

vecinos del elemento  $i$  cuya cardinalidad viene dada por  $N_i$ , por lo que la sumatoria es sobre todos los elementos  $j$  pertenecientes a la vecindad de  $i$ . El valor del índice de acoplamiento se encuentra en el rango de  $0 \leq \epsilon \leq 1$ . Cuando se tiene  $\epsilon = 0$  la Ec.(1.23) queda como una red de mapas sin acoplamiento, solamente con la dinámica local Ec.(1.24), que evoluciona independientemente de la conectividad

$$x_{t+1}(i) = f(x_t(i)) . \quad (1.24)$$

A medida que  $\epsilon$  se acerca a 1 el acoplamiento se hace más fuerte. Cuanto mayor es el valor de  $\epsilon$  mayor es la contribución de los vértices enlazados a  $v_i$  a la vez que disminuye el efecto de su propio valor para la siguiente iteración.

Si la red es completamente conexa la Ec.(1.23) queda de la siguiente forma:

$$x_{t+1}(i) = (1 - \epsilon)f(x_t(i)) + \frac{\epsilon}{N} \sum_{j=1}^N f(x_t(j)) , \quad (1.25)$$

que se conoce como red de mapas acoplados globalmente [4, 5, 46, 47, 48]. La sumatoria en este caso es sobre todos los nodos de la red. Si la red, en cambio, es lineal y se conectara con sus dos inmediatos vecinos queda de la siguiente forma:

$$x_{t+1}(i) = (1 - \epsilon)f(x_t(i)) + \frac{\epsilon}{2}(f(x_t(i+1)) + f(x_t(i-1))) . \quad (1.26)$$

Como se ve se puede modificar la red de mapas acoplados Ec.(1.23) de la forma que más convenga en cada caso.

### 1.4.1. Sincronización

Dependiendo de la conectividad, el acoplamiento y de la dinámica local, la red puede llegar a sincronizarse. El efecto es parecido a la sincronización en los péndulos de Huygens. Esto ocurrió en febrero de 1665 cuando Christiaan Huygens estaba convaleciente de gripe. Observó que los péndulos de dos relojes de su invención oscilaban perfectamente sincronizados, a pesar de que la probabilidad de que ocurra al azar es muy baja. Huygens dedujo que las vibraciones que se transmitían por la pared

donde estaban colgados los relojes hacían que estos oscilaran a la vez. Para comprobarlo colgó uno de los relojes en el lado opuesto de la habitación y al poco tiempo se desincronizaron los relojes. El acoplamiento de los relojes a través de la pared generaba la sincronización. En nuestro caso podemos considerar los nodos de la red como osciladores no lineales que interactúan entre sí dependiendo de la conectividad.

La ecuación Ec.(1.23) se puede poner en forma vectorial de la siguiente forma:

$$\mathbf{x}_{t+1} = \left[ (1 - \epsilon) \mathbf{I} + \frac{\epsilon}{v_i} \mathbf{M} \right] \mathbf{f}(\mathbf{x}_t), \quad (1.27)$$

donde se tiene que  $\mathbf{x}_t$  tiene componentes  $[\mathbf{x}_t]_i = x_t(i)$  y  $\mathbf{f}(\mathbf{x}_t)$  tiene componentes  $[\mathbf{f}(\mathbf{x}_t)]_i = f(x_t(i))$ ;  $\mathbf{M}$  es la matriz de adyacencia de dimensión  $N \times N$  e  $\mathbf{I}$  es una matriz identidad de dimensión  $N \times N$ .

La red acoplada se dice que presenta sincronización si

$$x_t(1) = x_t(2) = \dots = x_t(N) \rightarrow s_t, \quad (1.28)$$

cuando  $t \rightarrow \infty$ , donde  $s_t$  puede ser un punto de equilibrio, una órbita periódica e incluso una órbita caótica.

Del análisis de estabilidad lineal de estados sincronizados en redes de mapas acoplados, puede ser demostrado que estos estados son estables si se satisface la siguiente condición [6, 46]

$$\left| \left( 1 - \epsilon + \frac{\epsilon}{v_i} \mu_k \right) e^\lambda \right| < 1, \quad (1.29)$$

donde  $\{\mu_k : k = 1, \dots, N\}$  es el grupo de autovalores de la matrix de acoplamiento  $M$  y  $\lambda$  es el exponente de Lyapunov del mapa logarítmico.

Una forma de analizar el grado de sincronización de la red es calculando la desviación estándar del sistema. Ésta es una medida de dispersión que nos informa de la media de las distancias que tienen un conjunto de datos de una población respecto a su media aritmética, suele representarse por una  $S$  o con la letra  $\sigma$ . Dicho de otra manera es una medida del grado de dispersión de los datos del valor promedio. Se tiene que la desviación estándar está definida como

$$\sigma_t = \sqrt{\frac{\sum_{i=1}^N \bar{x}(t) - x_t(i)}{N}}, \quad (1.30)$$

donde,  $\bar{x}(t) = \sum_{i=1}^N x_t(i)$ , es el promedio instantáneo de los estados de los nodos del sistema.

Una desviación estándar grande indica que los datos están alejados del valor promedio, sus valores están dispersos y por lo tanto la red no está sincronizada. Una desviación estándar pequeña indica que la mayoría de los datos se encuentran cercanos al valor promedio, por lo tanto que sus valores difieren poco y la red se encuentra sincronizada.

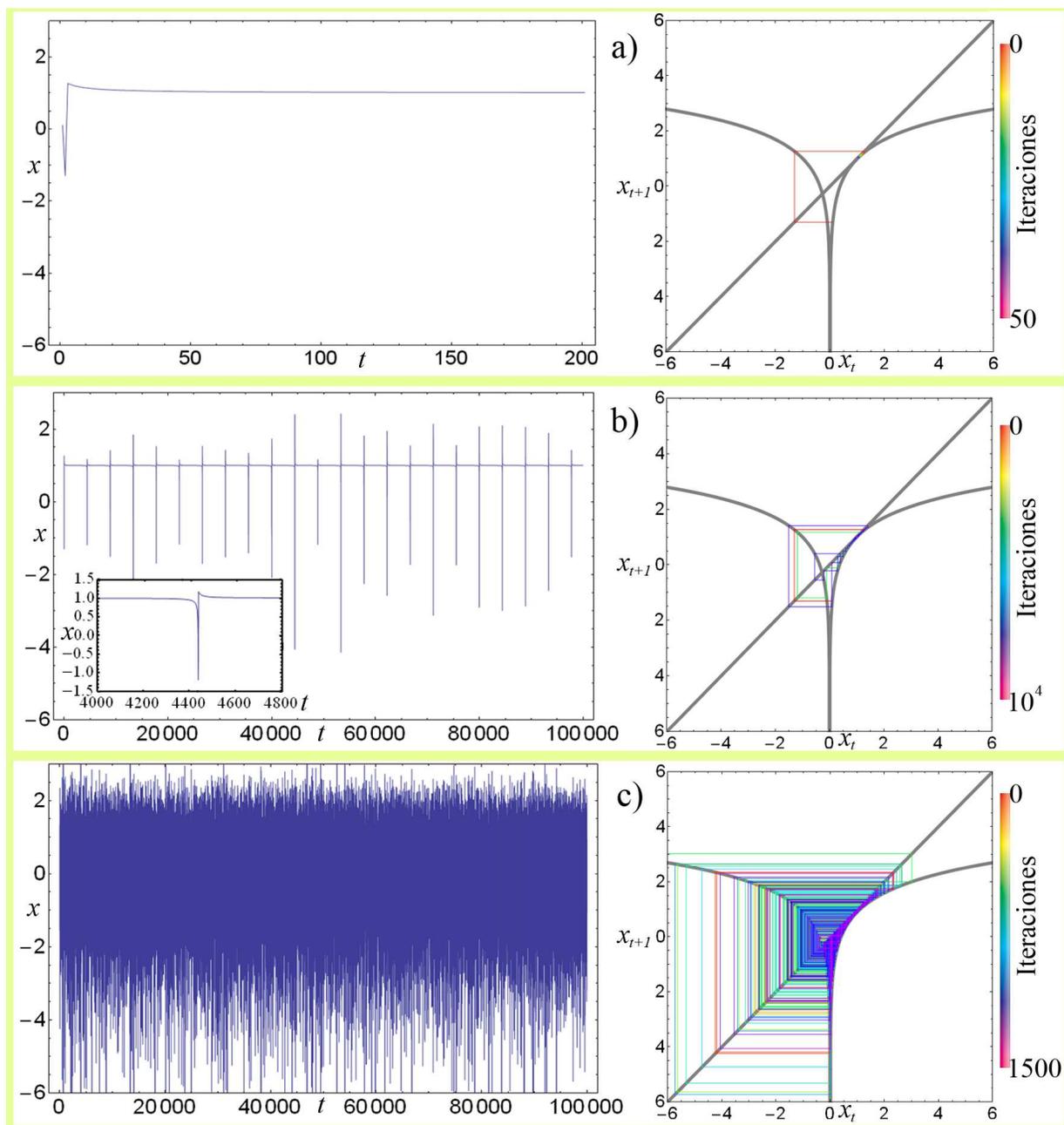


Figura 1.29: Comportamiento de  $x$  en el tiempo a la izquierda y la órbita que describe a la derecha para distintos valores de  $r$  del mapa logarítmico con  $x_0 = 0.1$ . En a) se tiene una órbita estable con  $r_{2c} = 1$ . En b) se observa el fenómeno de intermitencia tipo I con  $r = 0.999999$ . Se muestra una ampliación de una intermitencia que muestra la forma de la parte estable hacia el caos intermitente. En c) se encuentra el fenómeno de caos con  $r = 0.9$ .

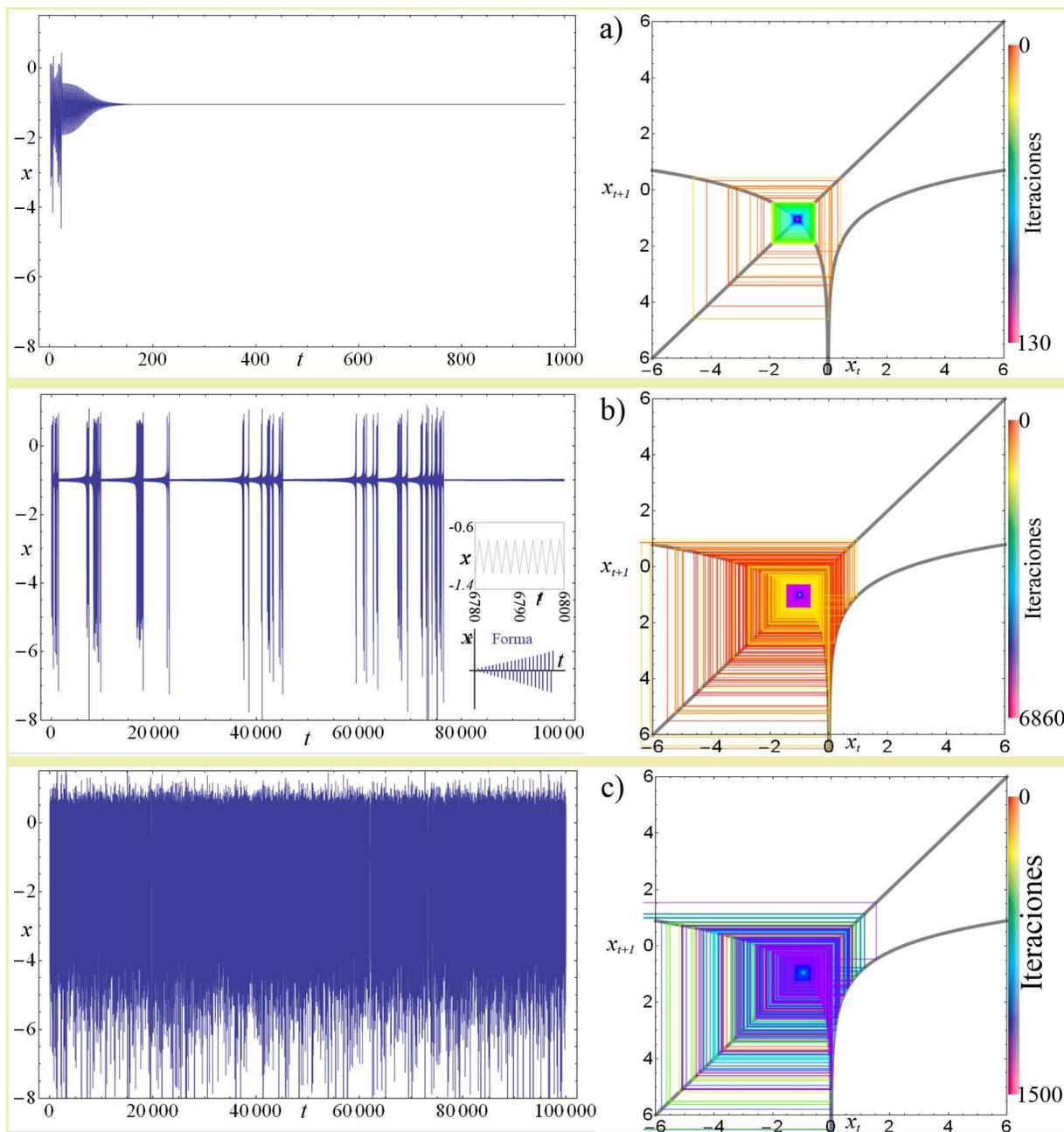


Figura 1.30: Comportamiento de  $x$  en el tiempo a la izquierda y la órbita que describe a la derecha para distintos valores de  $r$  del mapa logarítmico con  $x_0 = 0,1$ . En a) se tiene una órbita estable con  $r = -1,09$ . En b) se observa el fenómeno de intermitencia tipo III con  $r = -0,999985$ . Se muestra una ampliación que muestra la duplicación del período que aumenta de amplitud. En c) se encuentra el fenómeno de caos con  $r = -0,9$ .

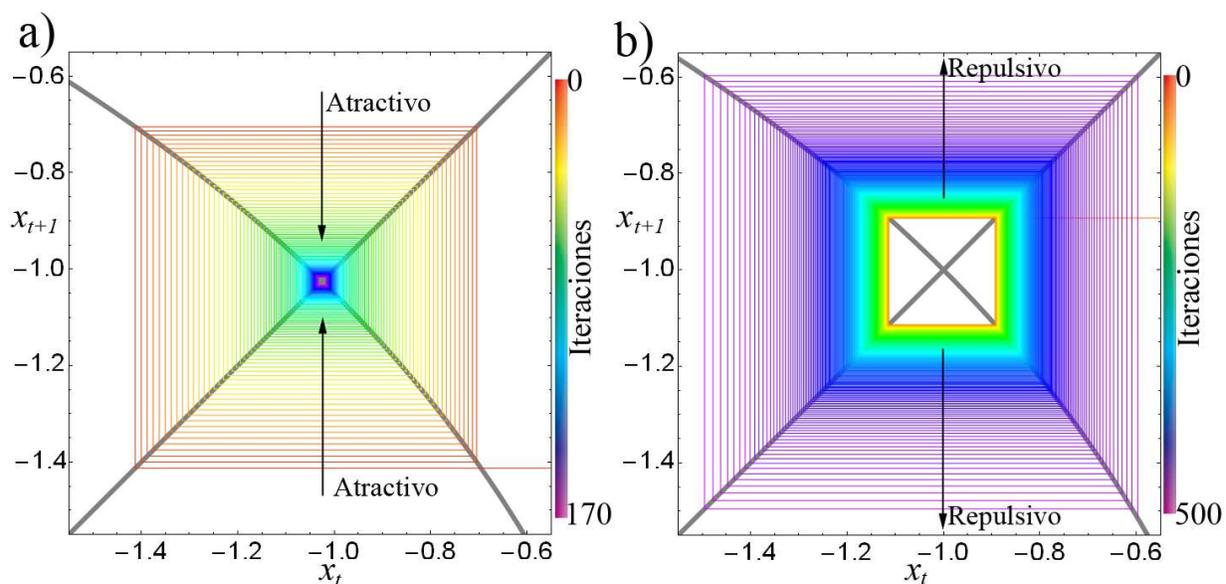


Figura 1.31: Acercamiento de la órbita del mapa logarítmico en la región donde se produce la intermitencia tipo III para distintos valores de  $r$ . En a) se tiene  $r = -1,05$  con  $\lambda < 0$  y en b) se tiene  $r = -0,9999$  con  $\lambda > 0$ .

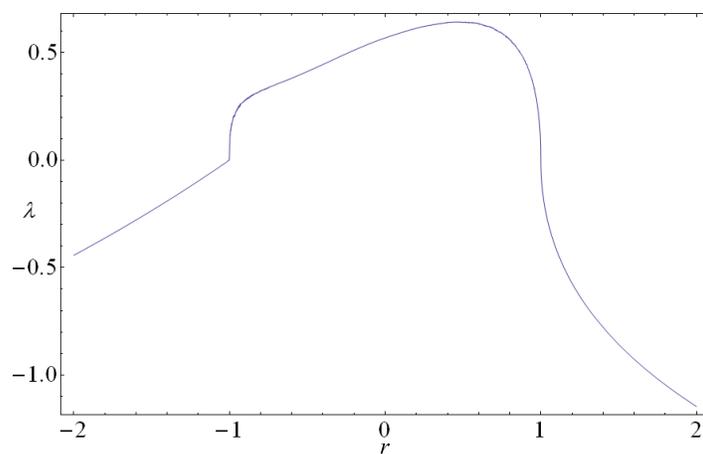


Figura 1.32: Exponente de Lyapunov del mapa logarítmico como función de  $r$  calculado sobre 100000 iteraciones por cada valor de  $r$ .

# Capítulo 2

## Metodología y resultados.

### 2.1. Metodología

Nuestra meta es encontrar una red cuya conectividad maximice la sincronización de la red con un mínimo de enlaces. Mediante un algoritmo genético minimizaremos dos funciones que reflejan la sincronización de una red de mapas acoplado y que tomarán en cuenta la desviación estándar y el promedio de los grados de los vértices. Cuando la población converja se analizarán los mejores individuos en busca de características comunes.

Para este trabajo se usa una red de mapas acoplados con acoplamiento difusivo, compuesta por  $N = 5000$  nodos, y definida por:

$$x_{t+1}(i) = (1 - \epsilon)f(x_t(i)) + \frac{\epsilon}{N_i} \sum_{j \in \nu_i} f(x_t(j)) , \quad (2.1)$$

donde,  $x_t(i)$  es el estado del elemento  $i$  en el tiempo discreto  $t$ ,  $i$  indexa los nodos con  $i = 1, 2, \dots, N$ ,  $\epsilon$  representa el coeficiente de difusión,  $f(x)$  es un mapa no lineal caótico y representa la dinámica local de cada nodo,  $\nu_i$  es el conjunto de nodos vecinos del elemento  $i$  cuya cardinalidad viene dada por  $N_i$ , por lo que la sumatoria es sobre

todos los elementos  $j$  pertenecientes a la vecindad de  $i$ . La fuerza de acoplamiento se fijó en  $\epsilon = 0,41$ , de forma tal que los individuos poco aptos no llegasen a sincronizarse.

El mapa  $f(x)$  que se usa es el logarítmico, definido como:

$$x_{t+1} = f(x_t) = b + \ln |x_t| , \quad (2.2)$$

donde,  $b$  es el parámetro de control. La elección del mapa logarítmico de basa en la dificultad de sincronizarlos ya que en la región caótica,  $b \in (-1, 1)$ , no presentan ventanas periódicas.

El estado sincronizado en el tiempo  $t$  se define como  $x_t(i) = x_t(j) \forall i, j$ . La sincronización de los elementos de la red puede ser caracterizada mediante el promedio temporal  $\langle \sigma \rangle$  de la desviación estándar instantánea de los estados del sistema

$$\sigma_t = \sqrt{\frac{\sum_{i=1}^N \bar{x}(t) - x_t(i)}{N}} , \quad (2.3)$$

donde,  $\bar{x}(t) = \sum_{i=1}^N x_t(i)$ , es el promedio instantáneo de los estados de los nodos del sistema.

Para el proceso de optimización se define como el genotipo del individuo a la matriz de adyacencia del grafo, de tamaño  $N \times N$ , definida como

$$M_{ij} = \begin{cases} 1 & , \text{ si } j \in \nu_i \\ 0 & , \text{ de lo contrario} \end{cases} , \forall i, j . \quad (2.4)$$

El tipo de grafos utilizados en este trabajo son no dirigidos y sin bucles por lo que la matriz de adyacencia debe cumplir con  $M_{ij} = M_{ji}$  y  $M_{ii} = 0$  respectivamente.

La optimización de la población de las redes de mapas acoplados se lleva a cabo mediante un algoritmo genético que parte con una población inicial de redes de compuesta por: ochenta redes aleatorias con un grado de vértice promedio  $5 \leq \bar{k} \leq 15$  [28, 29, 30]; cinco redes regulares de dimensión  $d = 1, 2, \dots, 5$  y grado de vecindad  $\kappa = 1, 2, \dots, 5$ ; cinco redes small world con dimensión del sustrato inicial  $d = 1, 2, \dots, 5$ , grado de vecindad inicial  $\kappa = 1, 2, \dots, 5$  y probabilidad de reconexión  $p \in [0, 1]$  [32], cinco redes tipo árbol con ramificación  $r = 1, 2, \dots, 10$  [15] y cinco redes fractales tipo triángulo de Sierpinski de dimensión  $d = 1, 2, \dots, 5$  [12]; para un

total de 100 individuos. El tamaño de las redes es  $N = 5000$  nodos, por lo que el genotipo de cada individuo está compuesto por  $N^2 = 25 \times 10^6$  genes.

Con cada individuo se simula la dinámica de la red de mapas acoplados definida por su matriz de adyacencia utilizando el software ISyS, *Inomogeneous System Simulator* [49]. Luego se evalúa el desempeño del individuo usando la función objetivo  $\phi$  y se actualizan los criterios de parada del algoritmo genético. De no cumplirse los criterios de parada, se crea una nueva generación mediante a aplicación de operadores genéticos de recombinación y mutación a la población actual. Esto último se repite hasta que se cumpla el criterio de parada.

La nueva generación está conformada por: los diez mejores individuos, diez individuos provenientes de la mutación de los diez mejores, diez individuos provenientes de la recombinación de los diez mejores con diez redes aleatorias que no sean las diez peores, treinta y cinco individuos que resultan de la recombinación aleatoria entre los ochenta individuos que no sean lo diez mejores ni los diez peores, cinco nuevos individuos con red aleatoria, cinco con red regular, cinco con red small world, cinco con red tipo árbol y cinco con red fractal. Los parámetros usados para nuevos individuos son los mismos que los usados en la creación de la población inicial.

Como el tamaño de las redes tipo árbol y fractales depende de sus parámetros de construcción es necesario ajustar el algoritmo de construcción de la matriz de adyacencia para que esta represente un grafo de tamaño  $N = 5000$ . Para ello se trunca el algoritmo de construcción obteniendo grafos como los que se muestran en la figura 2.1.

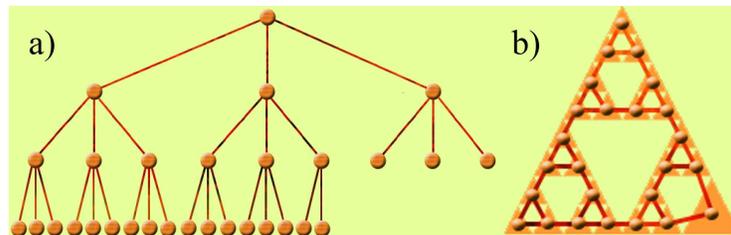


Figura 2.1: a) Red de 30 nodos con una estructura de árbol 3 regular. b) Red de 25 nodos en una estructura fractal de triángulo de Sierpinsky de dos dimensiones.

Con el algoritmo genético se plantean dos búsquedas minimizando la función objetivo.

La primera utiliza la función objetivo

$$\phi_1 = \langle \sigma \rangle . \quad (2.5)$$

Nótese que esta función no limita el número de enlaces de la red lo que permite llegar eventualmente a una red globalmente acoplada [46, 47, 48].

Para evitar el aumento indiscriminado del número de enlaces del grafo se plantea una segunda búsqueda utilizando la función objetivo

$$\phi_2 = \langle \sigma \rangle + (\bar{k} \times 10^{-5}) . \quad (2.6)$$

En la función objetivo  $\phi_2$  el peso del número promedio de enlaces por nodo  $\bar{k}$ , comienza a ser significativo una vez que el sistema sincronice, es decir cuando  $\langle \sigma \rangle < 10^{-5}$ .

La nueva generación está conformada en parte por individuos resultantes de mutaciones y recombinaciones hechas sobre individuos de la población precedente. El operador de recombinación, que actúa sobre dos individuos representados por sus respectivas matrices de adyacencia  $A$  y  $B$  crea un nuevo individuo  $C$ . Para ello el operador crea el conjunto  $V_A$  con  $N/2$  nodos del padre  $A$  seleccionados aleatoriamente y otro conjunto  $V_B$  con los  $N/2$  restantes. De esta forma se tiene que  $V_A \cap V_B = \emptyset$  y  $V_A \cup V_B = 1, 2, \dots, N$ . La matriz de adyacencia  $C$  del hijo de  $A$  y  $B$  se define en dos pasos, en el primero se establecen los enlaces definidos por uno de los padres, es decir cuando  $i, j \in V_M$  con  $M = A$  o  $B$ , como

$$C_{i,j} = \begin{cases} A_{i,j} & , \text{ si } i, j \in V_A \\ B_{i,j} & , \text{ si } i, j \in V_B \end{cases} . \quad (2.7)$$

El segundo asigna los enlaces que conectan a los nodos de  $V_A$  con los de  $V_B$ , es decir cuando  $i \in V_M$  y  $j \notin V_M$ , de forma tal que  $C_{ij} = A_{ij}$  o  $C_{ij} = B_{ij}$  con igual probabilidad. La figura 2.2 muestra un ejemplo de recombinación usando este operador.

Por último, para evitar que el algoritmo genético quede atrapado en un mínimo local, el 5% de los individuos que provienen del operador de recombinación sufre una mutación. A los individuos elegidos se les muta el 1% de sus genes, seleccionados de forma aleatoria, cambiando su estado; es decir que se añade un enlace entre dos

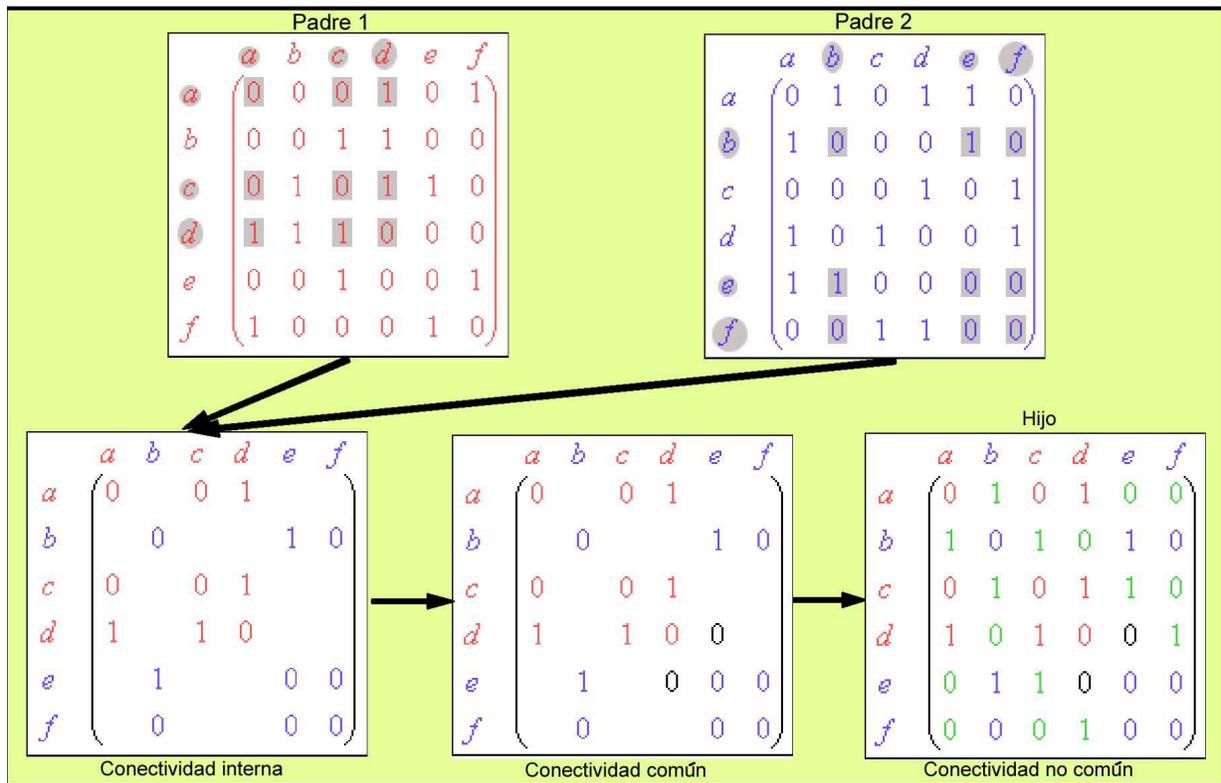


Figura 2.2: Proceso de recombinación. Tenemos en rojo y azul los vértices de cada padre, dentro de cuadros grises los vértices elegidos en cada padre, en color negro la conectividad del hijo, entre nodos pertenecientes a grupos distintos, que tiene igual conectividad en los dos padres y en verde la conectividad al azar del hijo, entre nodos pertenecientes a grupos distintos, donde los padres tienen conectividad no común.

nodos desconectados o se elimina un enlace entre nodos conectados. Este cambio se representa en la matriz de adyacencia sustituyendo un 0 por un 1 o un 1 por un 0 respectivamente. Esta mutación no aplica a los elementos en la diagonal de la matriz de adyacencia,  $M_{ii}$  y como la matriz de adyacencia es simétrica al mutar a la componente  $M_{ij}$  también se muta la componente  $M_{ji}$ .

Con este conjunto de parámetros y condiciones, el algoritmo genético converge antes de las 100 generaciones. Con el fin de caracterizar la topología de los mejores grafos [24], a los 10 mejores individuos de la última generación se les medirán el coeficiente de agrupamiento, la longitud característica y el grado de sus vértices.

El coeficiente de agrupamiento caracteriza la presencia de ciclos simples de orden tres

y se define como

$$C = \frac{3N_{\Delta}}{N_3}, \quad (2.8)$$

donde, el número de ciclos simples de grado tres dentro de la red,  $N_{\Delta}$ , viene dado por

$$N_{\Delta} = \sum_{k>j>i} M_{ij}M_{ik}M_{jk}. \quad (2.9)$$

Por su parte, el triple conexo  $N_3$  es un grupo de tres vértices donde hay un vértice central que se conecta a los otros dos directamente por medio de un enlace y se define como

$$N_3 = \sum_{k>j>i} (M_{ij}M_{ik} + M_{ji}M_{jk} + M_{ki}M_{kj}). \quad (2.10)$$

En ambos casos las sumatorias son sobre el conjunto de vértices diferentes  $i$ ,  $j$  y  $k$ .

La longitud característica es el promedio de la longitud de los caminos más cortos entre todos los posibles pares de vértices y se define como

$$l = \frac{1}{N(N-1)} \sum_{i \neq j} d_{ij}, \quad (2.11)$$

donde,  $d_{ij}$  es la longitud del camino más corto que conecta a los vértices  $i$  y  $j$ .

Por su parte, el grado de un vértice  $i$ ,  $k_i$ , es el número de enlaces incidentes a  $i$ . El número de enlaces que posee el vértice de mayor grado dentro de un grafo se denomina  $k_{max}$  y el número de enlaces que posee el vértice de menor grado en toda la red se denomina  $k_{min}$ . El promedio de los grados de todos los vértices viene dado por

$$\bar{k} = \sum_{i=1}^N \frac{k_i}{N}. \quad (2.12)$$

La distribución de grado,  $P(k)$ , indica la fracción de vértices de una red con grado  $k$ . Dependiendo del tipo de red la distribución puede ser de Poisson, exponencial, función delta y ley de potencias entre otros.

## 2.2. Resultados

En la figura 2.3 se muestra el progreso generacional del mejor individuo y de la media de los diez mejores individuos utilizando la función objetivo  $\phi = \langle \sigma \rangle$ . En ella se

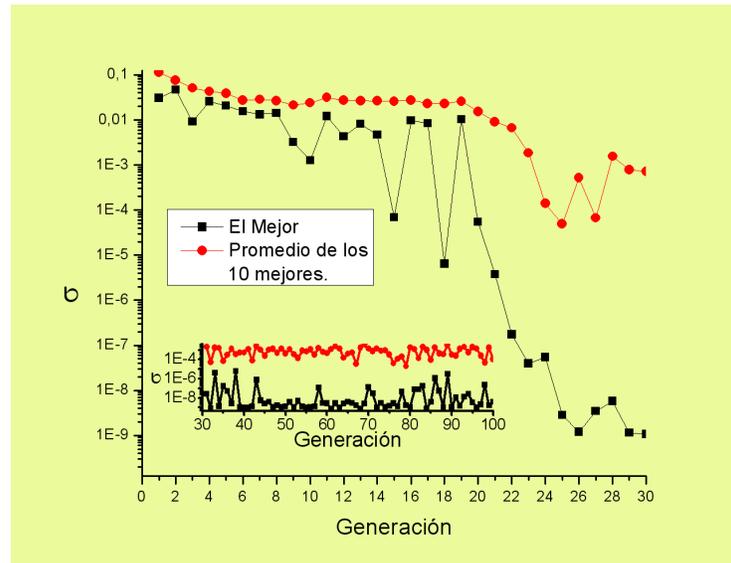


Figura 2.3: Progreso generacional de la función de evaluación  $\phi = \langle \sigma \rangle$  del individuo más apto y de la media de los diez primeros. En un cuadro más pequeño se muestra el comportamiento de la función de evaluación en el área de convergencia.

observa una rápida tendencia a la convergencia tanto del mejor individuo como de la media de los diez primeros. La desviación estándar del mejor individuo converge en la generación 26, a partir de este punto se puede decir que el promedio de la desviación estándar instantánea del mejor individuo es  $\langle \sigma \rangle \approx 10^{-8}$ , con oscilaciones alrededor de este valor<sup>1</sup>. Por su parte la media de la desviación estándar promedio de los diez mejores individuos se mantiene acotada  $\langle \sigma_{10} \rangle < 10^{-3}$  a partir de la generación 25.

En la figura 2.4 también se muestra el progreso generacional, pero esta vez utilizando la función de evaluación  $\phi_2 = \langle \sigma \rangle + (\bar{k} \times 10^{-5})$ . En este caso la convergencia es más rápida que en el caso anterior. La función de evaluación del mejor individuo converge en la generación 13 a  $\phi_2 \approx 10^{-4}$ , e igual que en el caso anterior se observan oscilaciones alrededor de este valor. Del mismo modo, la media de los diez mejores converge a  $10^{-2}$ .

En la figura 2.5 podemos ver los valores de longitud característica y el coeficiente de agrupamiento de las diez mejores redes resultantes al aplicar la primera función objetivo,  $\phi_1$ . Se puede apreciar que para las 10 redes los valores del coeficiente de

<sup>1</sup>El valor de  $\langle \sigma \rangle$  es levemente mayor que cero debido a errores numéricos.

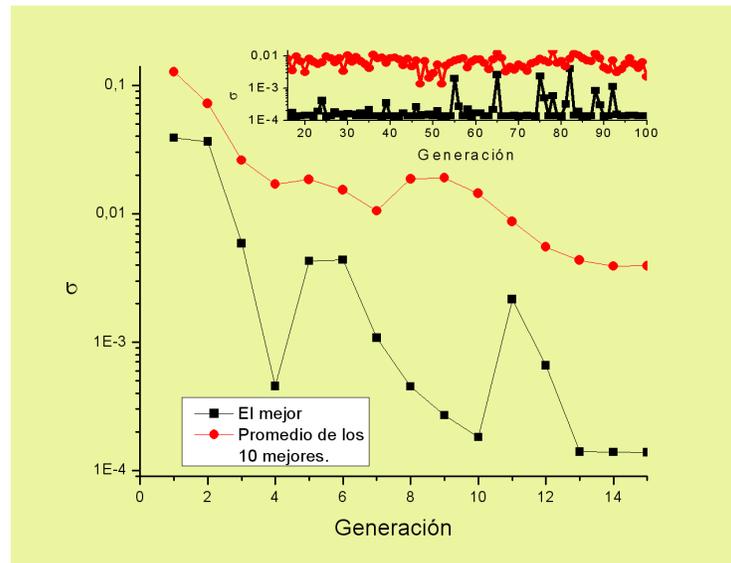


Figura 2.4: Progreso generacional de la función de evaluación  $f_2 = \sigma + (\bar{k}\Delta 10^{-5})$  del individuo más apto y de la media de los diez primeros. En un cuadro más pequeño se muestra el comportamiento de la función de evaluación en el área de convergencia.

agrupamiento es  $\mathcal{C} \ll 1$  y también la longitud característica es  $l \ll N$ .

Las medidas del coeficiente de agrupamiento y la longitud características de las 10 mejores redes resultantes de la aplicación de la función objetivo  $\phi_2$  se muestran en la figura 2.6. Al igual que en el caso anterior se tiene que  $\mathcal{C} \ll 1$  y  $l \ll N$ .

En ambos experimentos todas las redes resultantes son conexas. En las 10 mejores redes usando  $\phi_1$  el rango de valores de  $k_{min} \in [2, 4]$  y  $k_{max} \in [29, 34]$ . La figura 2.7 muestra la distribución de grados de los vértices de las 10 mejores redes obtenidas al usar la función objetivo  $\phi_1$ . Nótese que en las redes resultantes la distribución de grados de los vértices se ajusta a una distribución de Poisson. La figura también muestra como los valores del parámetro  $r$  de la distribución de Poisson se ajustan con los valores de  $\bar{k}$  tomados directamente de la red.

Al usar la segunda función objetivo,  $\phi_2$  se tiene que  $k_{min} \in [2, 4]$  y  $k_{max} \in [28, 31]$ . La figura 2.8 muestra que para el segundo experimento, la distribución de grados de los vértices también se ajusta a una distribución de Poisson y los valores del parámetro  $r$  se ajustan con los valores de  $\bar{k}$ .

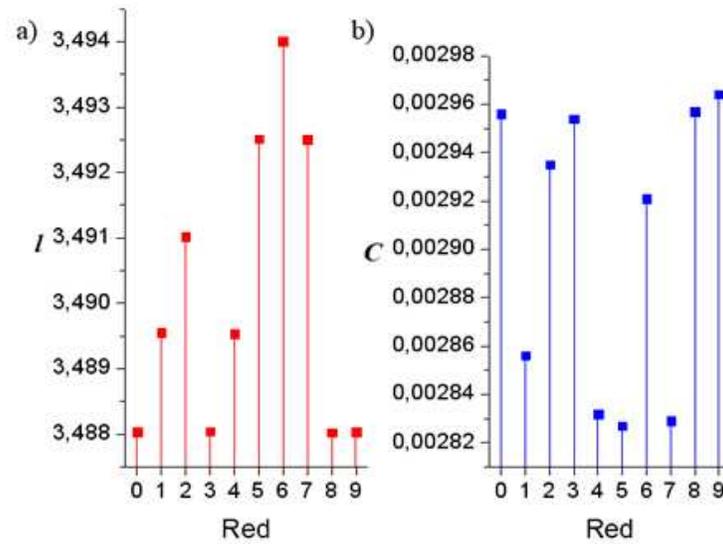


Figura 2.5: Valores de a) longitud característica y b) coeficiente de agrupamiento de las diez mejores redes resultantes para el primer experimento utilizando la función objetivo  $\phi_1 = \langle \sigma \rangle$ .

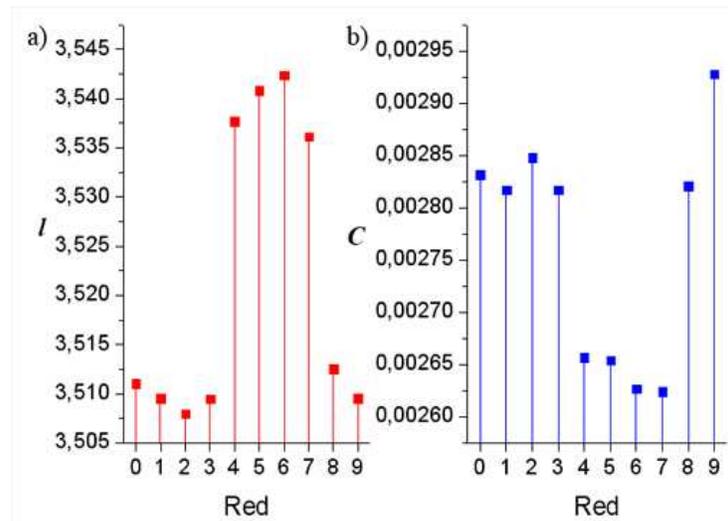


Figura 2.6: Valores de a) longitud característica y b) coeficiente de agrupamiento de las diez mejores redes resultantes para el segundo experimento utilizando la función objetivo  $\phi_2 = \langle \sigma \rangle + (\bar{k} \times 10^{-5})$ .

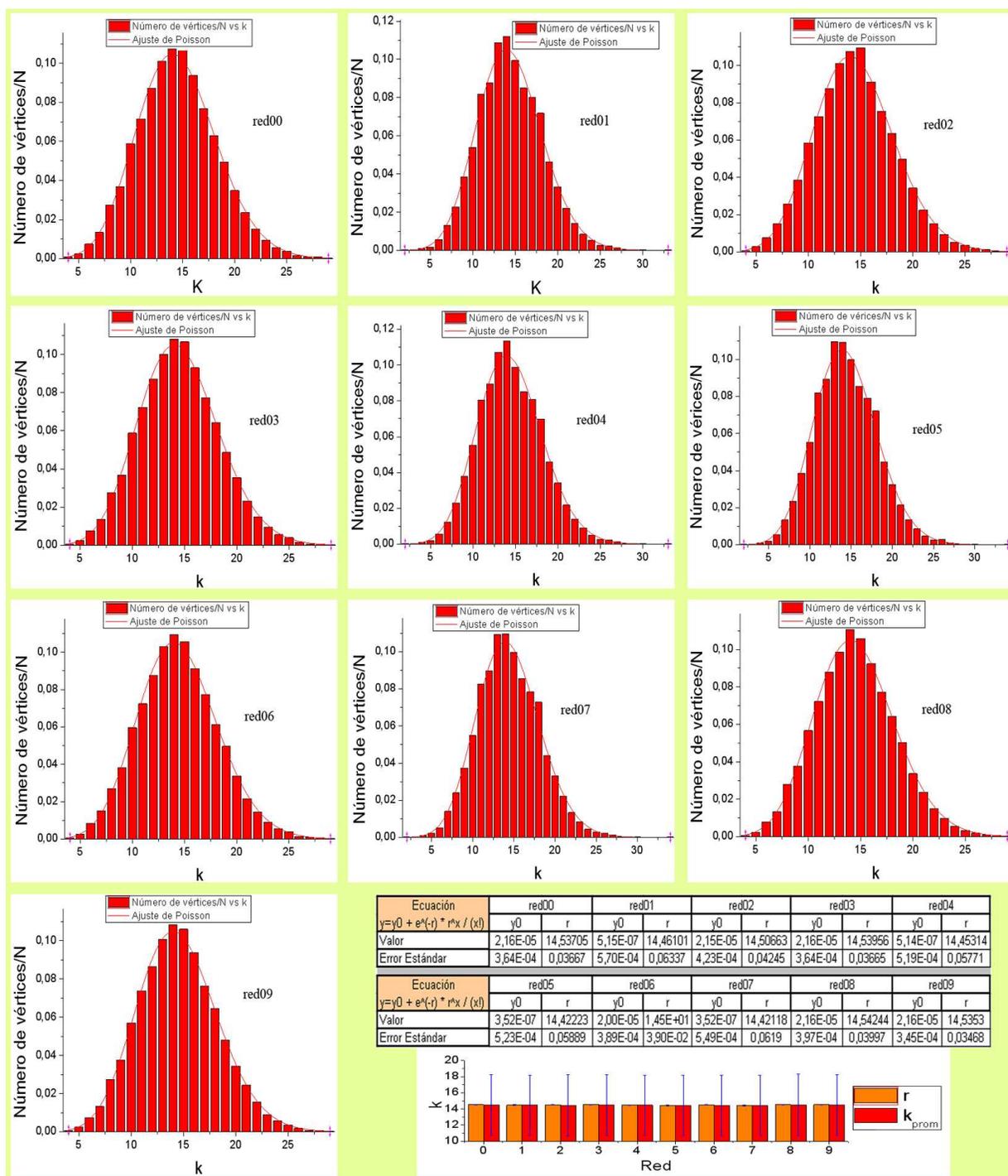


Figura 2.7: Distribución de grado de las redes resultantes del algoritmo genético para  $\phi_1 = \langle \sigma \rangle$ . Cada distribución tiene un ajuste de Poisson. En la esquina inferior derecha tenemos los valores de los parámetros del ajuste y la comparación del parámetro  $r$  del ajuste con el de  $\bar{k}$  tomado directamente de la red.

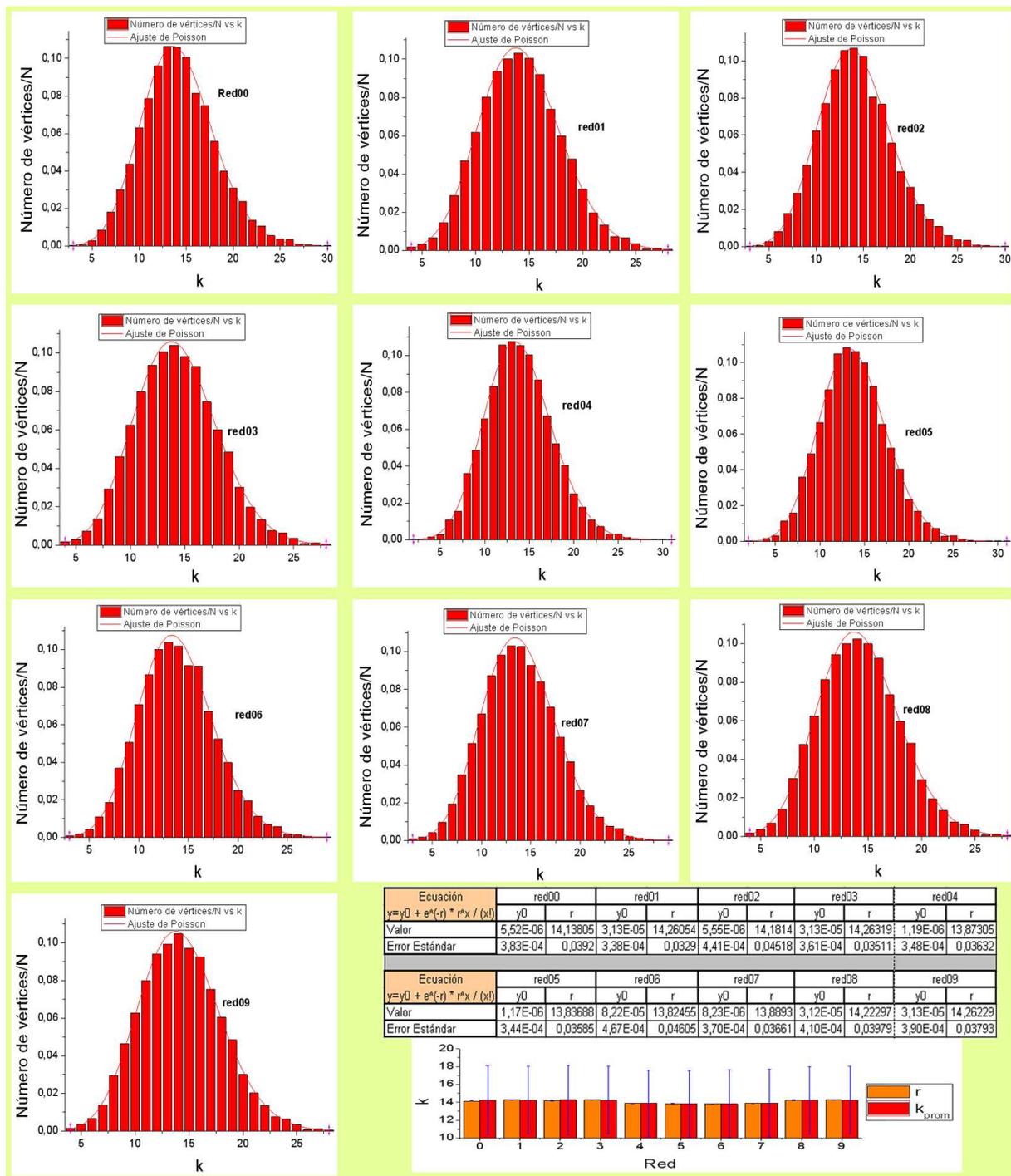


Figura 2.8: Distribución de grado de las redes resultantes del algoritmo genético para  $\phi_2 = \langle \sigma \rangle + (\bar{k} \times 10^{-5})$ . Cada distribución tiene un ajuste de Poisson. En la esquina inferior derecha tenemos los valores de los parámetros del ajuste y la comparación del parámetro  $r$  del ajuste con el de  $\bar{k}$  tomado directamente de la red.

# Capítulo 3

## Discusión de Resultados y Conclusiones.

### 3.1. Discusión de Resultados

El hecho de que las mejores redes que se obtienen al aplicar el algoritmo genético tengan longitud característica  $l$  pequeña, coeficiente de agrupamiento pequeño y una distribución de grados de los vértices que se ajusta a una distribución Poisson hace pensar que entre el conjunto de posibles redes complejas: Aleatorias, tipo Small World, libres de escala, etc.; y no complejas: regulares, tipo árbol, fractales, etc.; son las redes aleatorias tipo Erdős Rényi las que permiten que surja el fenómeno de sincronización con más facilidad. Esto por lo menos para el conjunto de parámetros utilizados en este trabajo.

En cuanto a la restricción que se introdujo a la función objetivo  $\phi_2$ , para evitar el crecimiento indiscriminado del grado de los vértices, se observa que a pesar que de hay un efecto en el nivel de sincronización que se obtiene en la red, no afecta sensiblemente las características topológicas que se midieron en este trabajo.

## 3.2. Conclusiones

El proceso de optimización mediante algoritmo genético utilizado en este trabajo dió como resultado redes cuya topología permite el fenómeno de sincronización en redes de mapas acoplados con más facilidad.

Con el conjunto de parámetros usados el estudio de las redes resultantes, por medio de la longitud característica, el coeficiente de agrupamiento y la distribución de grado de los vértices de cada red, las caracterizó claramente como redes aleatorias del modelo de Erdős-Rényi.

El uso de diferentes funciones objetivo altera ligeramente el valor de  $\bar{k}$  pero no el tipo de red resultante.

La investigación hecha en este trabajo puede ser enfocada en otras direcciones. Usando el mismo modelo de algoritmos genéticos se puede buscar las características topológicas de las redes que maximicen o minimicen otras propiedades colectivas emergentes como por ejemplo la formación de dominios, la separación de fases, el surgimiento de turbulencia, entre otras. También es posible utilizar otros caracterizadores de la topología de los grafos además de la longitud característica, el coeficiente de agrupamiento y la distribución de los grados de los vértices; como por ejemplo la vulnerabilidad, el grado de bipartividad, la dimensionalidad fractal, entre otros.

# Apéndice A

## Algoritmo Genético del Trabajo

```
program genetico
integer i, j, k, p, n, m, d, o, t, b(7000), r(3500), s(3500)
integer y(7000,7000), z(7000,7000), w(7000,7000), x(100)
integer q(49000000), gennum, gennum2, generacion
real eval, u(100), des, kpr
character(LEN=3) :: H
character(LEN=2) :: E, ind
character(LEN=1) :: A
c veo el .dat y los arreglo c voy a organizar las redes según su desviación estándar
write(*,*)
write(*,*) 'organizando las redes según su evaluación'
do i=1, 100
.  u(i)=0
.  x(i)=0
end do
open(2, file='eva.dat', status='old')
do i=1, 100
.  read(2,*) des, kpr
```

```

. eval= des
. p=0
. do j=1, 100
. . if (p .EQ. 0) then
. . . if (j .EQ. 100) then
. . . . if (eval .GT. u(100)) then
. . . . . do k=1, 99
. . . . . . u(k)=u(k+1)
. . . . . . x(k)=x(k+1)
. . . . . end do
. . . . . u(100)=eval
. . . . . x(100)=i-1
. . . . . p=1
. . . . . else
. . . . . do k=1, j-2
. . . . . . u(k)=u(k+1)
. . . . . . x(k)=x(k+1)
. . . . . end do
. . . . . u(j-1)=eval
. . . . . x(j-1)=i-1
. . . . . p=1
. . . . . end if
. . . . else
. . . . . if (eval .GT. u(j)) then
. . . . . . eval=eval
. . . . . else
. . . . . . do k=1, j-2
. . . . . . . u(k)=u(k+1)
. . . . . . . x(k)=x(k+1)
. . . . . . end do
. . . . . . u(j-1)=eval

```

```

. . . . . x(j-1)=i-1
. . . . . p=1
. . . . . end if
. . . end if
. . end if
. end do
end do
close(2)
open(3, file='gen.dat', status='old')
read(3,*) generacion
write(*,*)
write(*,*) 'generación=', generacion
close(3)
open(4, file='registro.dat', status='old')
if (generacion .EQ. 0) then
. do i=1, 10
. . write(4,*) i-1, ' ', u(i), ' ', x(i)
. end do
. write(4,*) '.....generación=', generacion
else
. do i=1, generacion*11
. . read(4,*) H
. end do
. do i=1, 10
. . write(4,*) i-1, ' ', u(i), ' ', x(i)
. end do
. write(4,*) '.....generacion=', generacion
end if
close(4)
do i=1, 10
. write(*,*) i-1, ' ', u(i), ' ', x(i)

```

```

end do
c conservacion de los 10 mejores
write(*,*) 'parte de los mejores'
do i=1, 10
.  if (x(i) .LT. 10) then
.  .  open(2, file='num.num', status='unknown')
.  .  write(2,*) x(i)
.  .  close(2)
.  .  open(2, file='num.num', status='old')
.  .  read(2,*) A
.  .  close(2)
.  .  ind='0'//A//''
.  else
.  .  open(2, file='num.num', status='unknown')
.  .  write(2,*) x(i)
.  .  close(2)
.  .  open(2, file='num.num', status='old')
.  .  read(2,*) E
.  .  close(2)
.  .  ind=''//E//''
.  end if
.  open(3, file=''//ind//'.ind', status='old')
.  read(3,*) n
.  do j=1, n*n
.  .  read(3,*) q(j)
.  end do
.  close(3)
.  E='00010203040506070809'(i*2-1:i*2)
.  open(4, file=''//E//'.new', status='unknown')
.  write(4,'(I4)') n
.  do j=1, n*n

```

```

. . write(4,'(I1)') q(j)
. end do
. close(4)
end do
cc
c mutacion de los 10 mejores
cc
write(*,*) 'mutación de los 10 mejores'
do i=1, 10
. if (x(i) .LT. 10) then
. . open(2, file='num.num', status='unknown')
. . write(2,*) x(i)
. . close(2)
. . open(2, file='num.num', status='old')
. . read(2,*) A
. . close(2)
. . ind='0'//A//''
. else
. . open(2, file='num.num', status='unknown')
. . write(2,*) x(i)
. . close(2)
. . open(2, file='num.num', status='old')
. . read(2,*) E
. . close(2)
. . ind=""//E//''
. end if
. open(3, file=""//ind//'.ind', status='old')
. read(3,*) n
. do j=1, n
. . do k=1, n
. . . read(3,*) y(k,j)

```

```

. . end do
. end do
. do o=1, 50
. . p=int(rand(0)*n)+1
. . m=int(rand(0)*n)+1
. . if (p .NE. m) then
. . . if (y(p,m) .EQ. 0) then
. . . . y(p,m)=1
. . . . y(m,p)=1
. . . else
. . . . y(p,m)=0
. . . . y(m,p)=0
. . . end if
. . end if
. end do
. close(3)
. E='10111213141516171819'(i*2-1:i*2)
. open(4, file="//E//".new', status='unknown')
. write(4,'(I4)' n
. do j=1, n
. . do k=1, n
. . . write(4,'(I1)' y(j,k)
. . end do
. end do
. close(4)
end do
c
c parte de 10 mejores con 10 cualquiera
c
write(*,*) 'recombinando los 10 mejores con 10 cualquiera'
do i=1, 10

```

```

. if (x(i) .LT. 10) then
. . open(2, file='num.num', status='unknown')
. . write(2,*) x(i)
. . close(2)
. . open(2, file='num.num', status='old')
. . read(2,*) A
. . close(2)
. . ind='0'//A//''
. else
. . open(2, file='num.num', status='unknown')
. . write(2,*) x(i)
. . close(2)
. . open(2, file='num.num', status='old')
. . read(2,*) E
. . close(2)
. . ind=''//E//''
. end if
. open(3, file=''//ind//'.ind', status='old')
. read(3,*) n
. do j=1, n
. . do k=1, n
. . . read(3,*) y(k,j)
. . end do
. end do
. close(3)
. m=int(rand(0)*80)+11
. if (x(m) .LT. 10) then
. . open(2, file='num.num', status='unknown')
. . write(2,*) x(m)
. . close(2)
. . open(2, file='num.num', status='old')

```

```

. . read(2,*) A
. . close(2)
. . ind='0'//A//''
. else
. . open(2, file='num.num', status='unknown')
. . write(2,*) x(m)
. . close(2)
. . open(2, file='num.num', status='old')
. . read(2,*) E
. . close(2)
. . ind=""//E//''
. end if
. open(4, file=""//ind//'.ind', status='old')
. read(4,*) n
. do j=1, n
. . do k=1, n
. . . read(4,*) z(k,j)
. . end do
. end do
. close(4)
c
cccccccccc aca separo los vertices que voy a agarrar de cada red
cccccccccc (hasta las otras cs), me van a quedar
cccccccccc dos arreglos de 50, r(j) y s(j), con los vértices separados c
. do j=1, n
. . b(j)=j
. end do
. t=n
. do j=1, n/2
10 . . if (t .NE. 2) then
. . . o=int(rand(0)*t)+1

```

```

. . . m=int(rand(0)*t)+1
. . . if (o .NE. m)then
. . . . r(j)=b(o)
. . . . s(j)=b(m)
. . . . if (o .EQ. t) then
. . . . . r(j)=b(o)
. . . . . else
. . . . . do k=o, t-1
. . . . . . b(k)= b(k+1)
. . . . . end do
. . . . . end if
. . . . if (m .EQ. t) then
. . . . . s(j)=b(m)
. . . . . t=t-2
. . . . . else
. . . . . if (o .LT. m) then
. . . . . . do k=m-1, t-2
. . . . . . . b(k)= b(k+1)
. . . . . . end do
. . . . . . t=t-2
. . . . . . else
. . . . . . do k=m, t-2
. . . . . . . b(k)= b(k+1)
. . . . . . end do
. . . . . . t=t-2
. . . . . . end if
. . . . . end if
. . . . else
. . . . . goto 10
. . . . end if
. . . else

```

```

. . . r(j)=b(1)
. . . s(j)=b(2)
. . end if
. end do
cccccccccccccccccccc
cahora voy a combinar las redes
c
cccccccccccccccccccc
. do j=1, n
. . do k=1, n
. . . w(k,j)=0
. . end do
. end do
. do j=1, n/2
. . do k=1, n/2
. . . w(r(k),r(j))=y(r(k),r(j))
. . . w(r(j),r(k))=w(r(k),r(j))
. . end do
. end do
. do j=1, n/2
. . do k=1, n/2
. . . w(s(k),s(j))=z(s(k),s(j))
. . . w(s(j),s(k))=w(s(k),s(j))
. . end do
. end do
. do j=1, n/2
. . do k=1, n/2
. . . if (y(r(k),s(j)) .EQ. z(r(k),s(j))) then
. . . . w(r(k),s(j))=y(r(k),s(j))
. . . . w(s(j),r(k))=w(r(k),s(j))
. . . else

```

```

. . . . w(r(k),s(j))=int(rand(0)*2)
. . . . w(s(j),r(k))=w(r(k),s(j))
. . . end if
. . end do
. end do
. E='20212223242526272829'(i*2-1:i*2)
. open(5, file="//E//'.new', status='unknown')
. write(5,'(I4)') n
. do j=1, n
. . do k=1, n
. . . write(5,'(I1)') w(j,k)
. . end do
. end do
. close(5)
end do

c
c ahora rcombino 50 hijos que no estén ni en los mejores ni en los peores
c
write(*,*) 'recombinando 45 hijos con 45 hijos'
do i=1, 45
. m=int(rand(0)*80)+11
. if (x(m) .LT. 10) then
. . open(2, file='num.num', status='unknown')
. . write(2,*) x(m)
. . close(2)
. . open(2, file='num.num', status='old')
. . read(2,*) A
. . close(2)
. . ind='0'//A//''
. else
. . open(2, file='num.num', status='unknown')

```

```

. . write(2,*) x(m)
. . close(2)
. . open(2, file='num.num', status='old')
. . read(2,*) E
. . close(2)
. . ind="//E//"
. end if
. open(3, file="//ind//'.ind', status='old')
. read(3,*) n
. do j=1, n
. . do k=1, n
. . . read(3,*) y(k,j)
. . end do
. end do
. close(3)
50 . d=int(rand(0)*80)+11
. if (d .EQ. m) then
. . goto 50
. end if
. if (x(d) .LT. 10) then
. . open(2, file='num.num', status='unknown')
. . write(2,*) x(d)
. . close(2)
. . open(2, file='num.num', status='old')
. . read(2,*) A
. . close(2)
. . ind='0'//A//"
. else
. . open(2, file='num.num', status='unknown')
. . write(2,*) x(d)
. . close(2)

```

```

. . open(2, file='num.num', status='old')
. . read(2,*) E
. . close(2)
. . ind="//E//"
. end if
. open(4, file="//ind//'.ind', status='old')
. read(4,*) n
. do j=1, n
. . do k=1, n
. . . read(4,*) z(k,j)
. . end do
. end do
. close(4)
cccccccccccccccccccc aca separo los vertices que voy a agarrar de cada red
cccccccccccccccccccc (hasta las otras cs), me van a quedar
cccccccccccccccccccc dos arreglos de 50, r(j) y s(j), con los vértices separados
. do j=1, n
. . b(j)=j
. end do
. t=n
. do j=1, n/2
20 . . if (t .NE. 2) then
. . . o=int(rand(0)*t)+1
. . . m=int(rand(0)*t)+1
. . . if (o .NE. m)then
. . . . r(j)=b(o)
. . . . s(j)=b(m)
. . . . if (o .EQ. t) then
. . . . . r(j)=b(o)
. . . . else
. . . . . do k=o, t-1

```

```

. . . . . b(k)= b(k+1)
. . . . . end do
. . . . . end if
. . . . . if (m .EQ. t) then
. . . . . . s(j)=b(m)
. . . . . . t=t-2
. . . . . else
. . . . . . if (o .LT. m) then
. . . . . . . do k=m-1, t-2
. . . . . . . . b(k)= b(k+1)
. . . . . . . end do
. . . . . . . t=t-2
. . . . . . else
. . . . . . . do k=m, t-2
. . . . . . . . b(k)= b(k+1)
. . . . . . . end do
. . . . . . . t=t-2
. . . . . . end if
. . . . . end if
. . . . . else
. . . . . goto 20
. . . . . end if
. . . . . else
. . . . . r(j)=b(1)
. . . . . s(j)=b(2)
. . . . . end if
. . . . . end do
cccccccccccccccccccc
cc ahora voy a combinar las redes
cccccccccccccccccccc
. . . . . do j=1, n

```

```

. . do k=1, n
. . . w(k,j)=0
. . end do
. end do
. do j=1, n/2
. . do k=1, n/2
. . . w(r(k),r(j))=y(r(k),r(j))
. . . w(r(j),r(k))=w(r(k),r(j))
. . end do
. end do
. do j=1, n/2
. . do k=1, n/2
. . . w(s(k),s(j))=z(s(k),s(j))
. . . w(s(j),s(k))=w(s(k),s(j))
. . end do
. end do
. do j=1, n/2
. . do k=1, n/2
. . . if (y(r(k),s(j)) .EQ. z(r(k),s(j))) then
. . . . w(r(k),s(j))=y(r(k),s(j))
. . . . w(s(j),r(k))=w(r(k),s(j))
. . . else
. . . . w(r(k),s(j))=int(rand(0)*2)
. . . . w(s(j),r(k))=w(r(k),s(j))
. . . end if
. . end do
. end do
. open(2, file='num.num', status='unknown')
. write(2,*) i+29
. close(2)
. open(2, file='num.num', status='old')

```

```

. read(2,*) E
. close(2)
. ind="//E//"
. open(5, file="//E//.new', status='unknown')
. write(5,'(I4)') n
. do j=1, n
. . do k=1, n
. . . write(5,'(I1)') w(j,k)
. . end do
. end do
. close(5)
end do
cc
c esta parte es para mutar redes que no sean los mejores
c ni los random ni los peores
cc
write(*,*) 'mutando las otras redes'
do i=20, 74
. p=int(rand(0)*100)+1
. if (p .LE. 5) then
. . open(2, file='num.num', status='unknown')
. . write(2,*) i
. . close(2)
. . open(2, file='num.num', status='old')
. . read(2,*) E
. . close(2)
. . ind="//E//"
. . open(3, file="//ind//.new', status='old')
. . read(3,*) n
. . do j=1, n
. . . do k=1, n

```

```

. . . . read(3,*) y(k,j)
. . . . end do
. . . end do
. . . do o=1, 50
. . . . p=int(rand(0)*n)+1
. . . . m=int(rand(0)*n)+1
. . . . if (p .NE. m) then
. . . . . if (y(p,m) .EQ. 0) then
. . . . . . y(p,m)=1
. . . . . . y(m,p)=1
. . . . . . else
. . . . . . y(p,m)=0
. . . . . . y(m,p)=0
. . . . . . end if
. . . . . end if
. . . . end do
. . . close(3)
. . . open(3, file="//ind//'.new', status='unknown')
. . . write(3,'(I4)') n
. . . do j=1, n
. . . . do k=1, n
. . . . . write(3,'(I1)') y(j,k)
. . . . . end do
. . . . end do
. . . end do
. . . close(3)
. . . end if
end do
write(*,*)
end

```

# Referencias

- [1] Fatihcan M. Atay, Türker Biyikoglu and Jürgen Jost, *On the synchronization of networks with prescribed degree distributions*, arXiv:nlin/0407024v2 [nlin.AO] 29 May 2005.
- [2] Fatihcan M. Atay, Jürgen Jost and Andreas Wende, *Delays, connection topology, and synchronization of coupled chaotic maps*, arXiv:cond-mat/0312177v2 [cond-mat.dis-nn] 11 Apr 2004.
- [3] Alex Arenas, Albert Diaz-Guilera and Conrad J. Perez-Vicente, *Synchronization Reveals Topological Scales in Complex Networks*, arXiv:cond-mat/0511730v2 [cond-mat.dis-nn] 1 Dec 2005.
- [4] K. Kaneko, *Period doubling of kink-antikink patterns, quasiperiodicity in antiferro-like structures and spatial intermittency in coupled map lattices*, Prog. Theor. Phys., 72 480–486, 1984.
- [5] K. Kaneko, *Spatial period-doubling in open flow*, Phys. Lett. A111 321– 325, 1985.
- [6] I. Waller and R. Kapral, *Spatial and temporal structure in systems of coupled nonlinear oscillators*, Phys. Rev., A30 2047–2055, 1984.
- [7] R. Kapral, *Pattern formation in two-dimensional arrays of coupled discrete time oscillators*, Phys. Rev. A31 3868–3879, 1985.
- [8] S. P. Kuznetsov, *On critical behavior of one-dimensional lattices*, Pis'ma J. Techn. Fiz. 9 , 94–98, 1983.

- [9] S. P. Kuznetsov, *On model description of coupled dynamical systems near the transition point order-disorder*, Izv. VUZov-Fizika 27 87–96, 1984.
- [10] O. Alvarez-Llamoza, K. Tucci, M. G. Cosenza, and M. Pineda, *Random global coupling induces synchronization and nontrivial collective behavior in networks of chaotic maps*, arXiv:nlin/0612010v1 [nlin.CD] 5 Dec 2006.
- [11] M. Barahona and L. M. Pecora, *Synchronization in small-world systems*, Phys. Rev. Lett, vol. 89, no. 5, 054101, 2002.
- [12] M.G. Cosenza, K. Tucci, *Turbulence in small-world networks*, arXiv:nlin/0112037v1 [nlin.CD] 20 Dec 2001.
- [13] Jesús Gómez-Gardenes, Yamir Moreno, Alex Arenas, *Paths to Synchronization on Complex Networks*, arXiv:cond-mat/0608314v3 [cond-mat.stat-mech] 31 Jan 2007.
- [14] H. Hong, M. Y. Choi, and Beom Jun Kim, *Synchronization on small-world networks*, arXiv:cond-mat/0110359v1 [cond-mat.dis-nn] 18 Oct 2001.
- [15] M. G. Cosenza and K. Tucci, *Pattern Formation on Trees*, arXiv:nlin/0105008v1 [nlin.PS] 4 May 2001.
- [16] K. Tucci, M. G. Cosenza, *Spectral properties and pattern selection in fractal growth networks*, arXiv:nlin/0312017v1 [nlin.PS] 8 Dec 2003.
- [17] Mandelbrot, B.B., *Les objets fractals, forme, hasard et dimension*, Paris: Flammarion, 1975.
- [18] Mandelbrot, B.B., *The Fractal Geometry of Nature*, W.H. Freeman and Company, 1982.
- [19] Luca Donetti, Pablo I. Hurtado, Miguel A. Munoz, *Network synchronization: Optimal and Pessimial Scale-Free Topologies*, arXiv:0710.4886v1 [cond-mat.dis-nn] 25 Oct 2007.
- [20] F. Sorrentino, M. di Bernardo, G. Huerta Cuellar, S. Boccaletti, *Synchronization in weighed scale-free networks with degree-degree correlation*, arXiv:cond-mat/0611653v1 [cond-mat.dis-nn] 26 Nov 2006.

- [21] Richard Johnsonbaugh. 1999. *Matemáticas discretas*. México. Prentice Hall.
- [22] Bondy y Murty. 1976. *Graph Theory with Applications*. Estados Unidos de Norte América. Elsevier Science Publishing Co., Inc.
- [23] Chris Godsil y Gordon Royle, *Algebraic Graph Theory*, Nueva York: Springer-Verlag. 2001.
- [24] L. da F. Costa, F. A. Rodrigues, G. Travieso y P. R. Villas Boas, *Characterization of Complex Networks: A Survey of measurements*, arXiv:cond-mat/0505185 v5 16 Aug 2006.
- [25] Roberto F. S. Andrade, José G. V. Miranda y Suani T. R. Pinho, *Characterization of complex networks by higher order neighborhood properties*, arXiv:cond-mat/0612695v1 29 Dec 2006.
- [26] M. E. J. Newman, *The structure and function of complex networks*, arXiv:cond-mat/0303516v1 [cond-mat.stat-mech] 25 Mar 2003.
- [27] Réka Albert y Albert-László Barabási, *Statistical Mechanics of Complex Networks*, arXiv:cond-mat/0106096v1 [cond-mat.stat-mech] 6 Jun 2001.
- [28] P. Erdős y A. Rényi, *On random graphs*, Publicationes Mathematicae, 6:290–297, 1959.
- [29] P. Erdős y A. Rényi, *On the evolution of random graphs*, Publ. Math. Inst. Hungar. Acad. Sci, 5:17–61, 1960.
- [30] P. Erdős y A. Rényi, *On the strenght of connectedness of a random graph*, Acta Mathematica Scientia Hungary, 12:261–267, 1961.
- [31] Gilbert E, *Random graphs*, Ann. Math. Stat. 1959;30:1141–1144.
- [32] Watts D, Strogatz S, *Collective dynamics of small-world networks*, Nature. 1998;393:440–442. doi:10.1038/30918.
- [33] Holland, John H (1975), *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor.
- [34] Goldberg, D.E. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, MA. 1989.

- [35] Kay Tucci. *Computación emergente*. <http://webdelprofesor.ula.ve/ciencias/kay/>.
- [36] Fogel David B., *Evolutionary Computation*. IEEE Press. Estados Unidos. 1995.
- [37] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, y Clifford Stein, *Introduction to Algorithms*, second edition. MIT Press and McGraw-Hill. 2001.
- [38] Goldberg, D.E. y Richardson J., *Genetic algorithm with sharing for multimodal function optimization*. Proceeding of the 2st ICGA. pp. 41-49, 1987.
- [39] W. de Melo y S. van Strien, *One-Dimensional Dynamics*, Springer-Verlag, 1993.
- [40] P. Collet y J. P. Eckmann, *Iterated Maps on the Interval as Dynamical Systems*, Birkhauser, 1980.
- [41] K. T. Alligood, T. D. Sauer, y J. A. Yorke, *Chaos: An Introduction to Dynamical Systems*, Springer-Verlag, New York, 1996.
- [42] Manneville, P., y Pomeau, Y., *Intermittency and the Lorenz Model*, Phys. Lett. 75A, 1, 1979.
- [43] Manneville, P., y Pomeau, Y., *Different Ways to Turbulence in Dissipative Dynamical Systems*, Physica 1D, 219, 1980.
- [44] Pomeau, Y., y Manneville, P., *Intermittent Transition to Turbulence in Dissipative Dynamical Systems*, Comm. Math. Phys. 74, 189, 1980.
- [45] Schuster, Heinz G., *Deterministic Chaos*, VCH Verlagsgesellschaft, 1989.
- [46] M. G. Cosenza y J. González, *Synchronization and Collective Behavior in Globally Coupled Logarithmic Maps*, Progress of Theoretical Physics Vol. 100 No. 1, 1998.
- [47] M. G. Cosenza y A. Parravano, *Dynamics of Coupling Functions in Globally Coupled Maps: Size, Periodicity and Stability of Clusters*, arXiv:nlin/0107046v1 [nlin.CD] 20 Jul 2001.
- [48] M. G. Cosenza and A. Parravano, *Turbulence in Globally Coupled Maps*, arXiv:chao-dyn/9603004v1 5 Mar 1996.
- [49] Kay Tucci, *Procesos Dinámicos espaciotemporales en Redes Inhomogéneas*, Tesis Doctoral, Universidad de los Andes. 2002.

