

# Fundamentos de Algoritmos Genéticos (AG).

Dr. Felipe Padilla Díaz<sup>1</sup>  
M.A. Francisco Álvarez Rodríguez<sup>2</sup>

## INTRODUCCIÓN

El objetivo de este material es dar los fundamentos básicos, terminología, desarrollos y resultados de los AG en general y su aplicación a problemas diversos. La exposición incluye la descripción de las componentes de un AG, su funcionamiento, el Teorema de esquemas, las modificaciones, extensiones y técnicas avanzadas, para tratar con más detalle los AG diseñados para problemas de secuenciación (PS). Se parte del análisis de la dificultad que presenta el AG simple para modelar este tipo de problemas y de cómo esto originó la aparición de diferentes representaciones del cromosoma. Se proponen diversas formas de agrupar las representaciones del cromosoma, teniendo en cuenta tres criterios que destacan las principales tendencias de la investigación en los PS. Tres aspectos importantes en el estudio del comportamiento de los AG diseñados para PS son analizados.

### 1.1 Componentes de un AG

Los AG surgen como herramientas para la solución de complejos problemas de optimización producto del análisis de los sistemas adaptativos en la naturaleza, y como resultado de abstraer la esencia de su funcionamiento.

Los AG son procedimientos de propósito general y de búsqueda paralela, basados en los principios de la genética y la evolución. Ellos trabajan modificando repetidamente una población de estructuras artificiales a través de la aplicación de los siguientes operadores genéticos:

- Operador de Selección o Darwiniano
- Operador de Entrecruzamiento o Mendeliano
- Operador de Mutación

La meta en optimización es encontrar la mejor solución posible o soluciones a un problema, con respecto a uno o más criterios. Para utilizar los AG es necesario encontrar una posible estructura para representar las soluciones. Pensando este asunto como el problema de buscar en un espacio de estados, una instancia de esta estructura representa un punto o un estado en el espacio de búsqueda de todas las posibles soluciones.

Así, una estructura de datos en el AG consistirá en uno o más cromosomas (frecuentemente uno). Un *cromosoma* es comúnmente una cadena de bits, sin embargo, otras representaciones del cromosoma pueden incluir vectores de números reales (Davis, 1991; Eshelman y Shaffer, 1993; Goldberg, 1991a, 1991b) y programas de computadoras de alto nivel (Koza, 1992). Habitualmente la estructura cromosomática tiene longitud fija  $\ell$ , sin embargo, se conocen problemas donde la longitud variable es más apropiada.

Cada cromosoma (cadena) es una concatenación de un número de subcomponentes

<sup>1</sup> Jefe del Departamento de Sistemas Electrónicos.  
E-mail: fpadilla2000@yahoo.com

<sup>2</sup> Coordinador de la Especialidad en Redes, Centro de Ciencias Básicas.

llamados *genes*. La posición de un gen en el cromosoma se conoce como *locus* y sus valores como *alelos*. En la representación como cadena de bits, un gen es un bit, un locus es su posición en la cadena y un alelo es su valor 0 ó 1. El término biológico *genotipo* se refiere a la estructura genética total de un individuo. El término *fenotipo* se refiere a las características observables de un individuo y corresponde a su estructura decodificada en el AG.

Tomemos un ejemplo sencillo para comprender lo hasta aquí expuesto. Sea el problema de optimización genética la maximización de la siguiente función de dos variables:

$$F(x_1, x_2) = x_1 + x_2$$

donde:  $0 \leq x_1 \leq 1$  y  $0 \leq x_2 \leq 1$ .

Una forma de codificación para las variables reales es transformarlas en cadenas de enteros binarios de una longitud apropiada para lograr la precisión deseada. Para nuestro ejemplo la codificación en 8-bits es suficiente para cada variable  $x_1$  y  $x_2$ . La decodificación de estas variables se hace dividiendo el entero binario correspondiente por  $2^7-1$ . Por ejemplo, 00000000 representa  $0/127$  ó 0, 11111111 representa  $127/127$  ó 1. La estructura de datos a optimizar es una cadena de 16 bits que representa la concatenación de las codificaciones de  $x_1$  y  $x_2$ . La variable  $x_1$  se ubica en los primeros 8 bits y la variable  $x_2$  se ubica en los 8 bits más a la izquierda. El genotipo de un individuo es una cadena de 16 bits, mientras que el fenotipo es una instancia del tuplo  $\langle x_1, x_2 \rangle$ . El genotipo es un punto en el espacio de Hamming de dimensión 16 donde el AG realiza su búsqueda. El fenotipo es un punto en el espacio dos dimensional de las variables decodificadas.

Al optimizar una estructura usando un AG se necesita una medida de la calidad de cada estructura en el espacio de búsqueda. La función de adaptabilidad es la encargada de esta tarea. En una maximización de funciones, la función objetivo frecuentemente actúa como la función de adaptabilidad, como en el ejemplo anterior, en el cual la meta es encontrar el valor de  $\langle x_1, x_2 \rangle$  que maximice  $F$ . Los AG realizan una maximización por defecto, para los problemas de minimización los valores de la función objetivo pueden ser negados con vistas a tomar valores positivos para producir así la adaptabilidad.

## 1.2 Funcionamiento de un AG

Los mecanismos de un AG simple son como sigue (Padilla 1998). El AG simple genera aleatoriamente una población de  $n$  estructuras (cadenas, cromosomas o individuos) y sobre ésta actúan los operadores transformando la población. Una vez completada la acción de los tres operadores se dice que ha transcurrido un ciclo generacional.

El operador de selección o Darwiniano realiza la selección de las cadenas de acuerdo a su adaptabilidad para el posterior apareamiento.

El operador de entrecruzamiento o Mendeliano realiza la recombinación del material genético de dos cadenas padres.

El operador de Mutación al estilo del operador natural realiza la mutación de un gen dentro de un cromosoma o cadena a sus diferentes formas aleomorfas.

Para cada uno de estos operadores está asociado el uso de probabilidades y la generación de números aleatorios.

El AG ejecuta para un número fijo de generaciones o hasta que se satisface algún criterio de parada. Durante cada generación el AG simple ejecuta en el siguiente orden, la selección proporcional a la adaptabilidad, el entrecruzamiento de un solo punto y la mutación.

La selección proporcional a la adaptabilidad se realiza de la siguiente manera:

Se asigna a cada individuo  $i$  de la población una probabilidad de selección  $P_s(i)$ , de acuerdo a la proporción entre la adaptabilidad de  $i$  y la adaptabilidad de la población total,

$$P_s(i) = f(i) / \sum_{j=1}^n f(j)$$

Entonces se seleccionan (con reemplazamiento) un total de  $n$  individuos para posteriores procesamientos genéticos de acuerdo a la distribución definida por  $P_s(i)$ . La forma más simple de selección proporcional es la ruleta (Goldberg, 1989a). Cada individuo tiene un sector en la ruleta proporcional a  $P_s(i)$ . Intuitivamente, la función de adaptabilidad, puede verse como una medida de ganancia, utilidad o ajuste que queremos maximizar.

Copiando cadenas de acuerdo a su adaptabilidad significa que las cadenas con valores más altos tienen mayores probabilidades de contribuir en uno o más hijos en la próxima generación. Cuando una cadena es seleccionada para reproducción, se hace una copia exacta de la misma y pasa a una bolsa de apareamiento, de donde actuarán próximos operadores. Este operador es una versión de la selección natural darwiniana de la supervivencia. En las poblaciones naturales, la adaptabilidad es una cierta habilidad de las criaturas a sobrevivir a los depredadores, la pestilencia y otros obstáculos hacia la madurez y consecuente reproducción. En el esquema artificial la función de adaptabilidad es el árbitro final de la vida o muerte de los individuos (cadenas).

Después de la selección, los  $n$  individuos que se encuentran en la bolsa de apareamiento son acoplados aleatoriamente para producir  $n/2$  parejas. Existe una probabilidad de entrecruzamiento fija  $P_c$ . Para cada pareja el entrecruzamiento puede ocurrir o no. Con probabilidad  $1 - P_c$  el entrecruzamiento no ocurre y ambos individuos pasan intactos a la mutación. De lo contrario la pareja produce dos hijos por el entrecruzamiento y sólo ellos pasan a la mutación.

La obtención de individuos nuevos a partir de los ya existentes es una de las características más interesantes e importantes en el trabajo de los AG.

Se toman dos cadenas  $B$  y  $C$  que van a ser entrecruzadas. Para poder hacer esto es necesario determinar el punto por donde se hará el cruce y después lo que queda es intercambiar información:

Después del entrecruzamiento comienza a efectuarse la mutación. Para todas las cadenas que llegan al estado de mutación, cada uno de sus bits son modificados con probabilidad  $P_m$ . Los bits de cada cadena son mutados independientemente, es decir, la mutación de un bit no afecta la probabilidad de mutación de los otros bits. El AG simple trata la mutación sólo como un operador secundario con el rol de restaurar la pérdida de material genético. Por ejemplo, supongamos que todas las cadenas en una población tienen valor 0 en una posición determinada y la solución óptima tiene un 1 en esa posición. El entrecruzamiento no puede regenerar un 1 en esa posición mientras que la mutación sí.

La población resultante de la mutación reemplaza totalmente la población anterior completando una generación. Las generaciones siguientes realizan el

mismo ciclo de selección, entrecruzamiento y mutación.

Se puede resumir en el esquema de la página siguiente, el funcionamiento de un AG simple:

Desde el punto de vista de la comparación de los AG con otros métodos de búsqueda se pueden enmarcar sus diferencias en cuatro aspectos:

- 1- Trabajan con una codificación de los parámetros y no con los parámetros mismos.
- 2- Buscan a partir de una población de puntos y no de un punto simple.
- 3- Usan directamente la función objetivo y no la derivada u otro conocimiento auxiliar.
- 4- Usan reglas de transición probabilísticas y no determinísticas.

### 1.3 Teorema de Esquema

Mientras un AG explícitamente procesa cadenas, implícitamente éste procesa esquemas, los cuales representan similitudes entre las cadenas (Goldberg, 1989a; Holland, 1992). El AG no puede proponerse visitar todos los puntos en el espacio de búsqueda, y sí muestrear un número suficientemente grande de hiperplanos en regiones de alta adaptabilidad del espacio de búsqueda. Cada hiperplano corresponde a un conjunto de subcadenas similares de alta adaptabilidad.

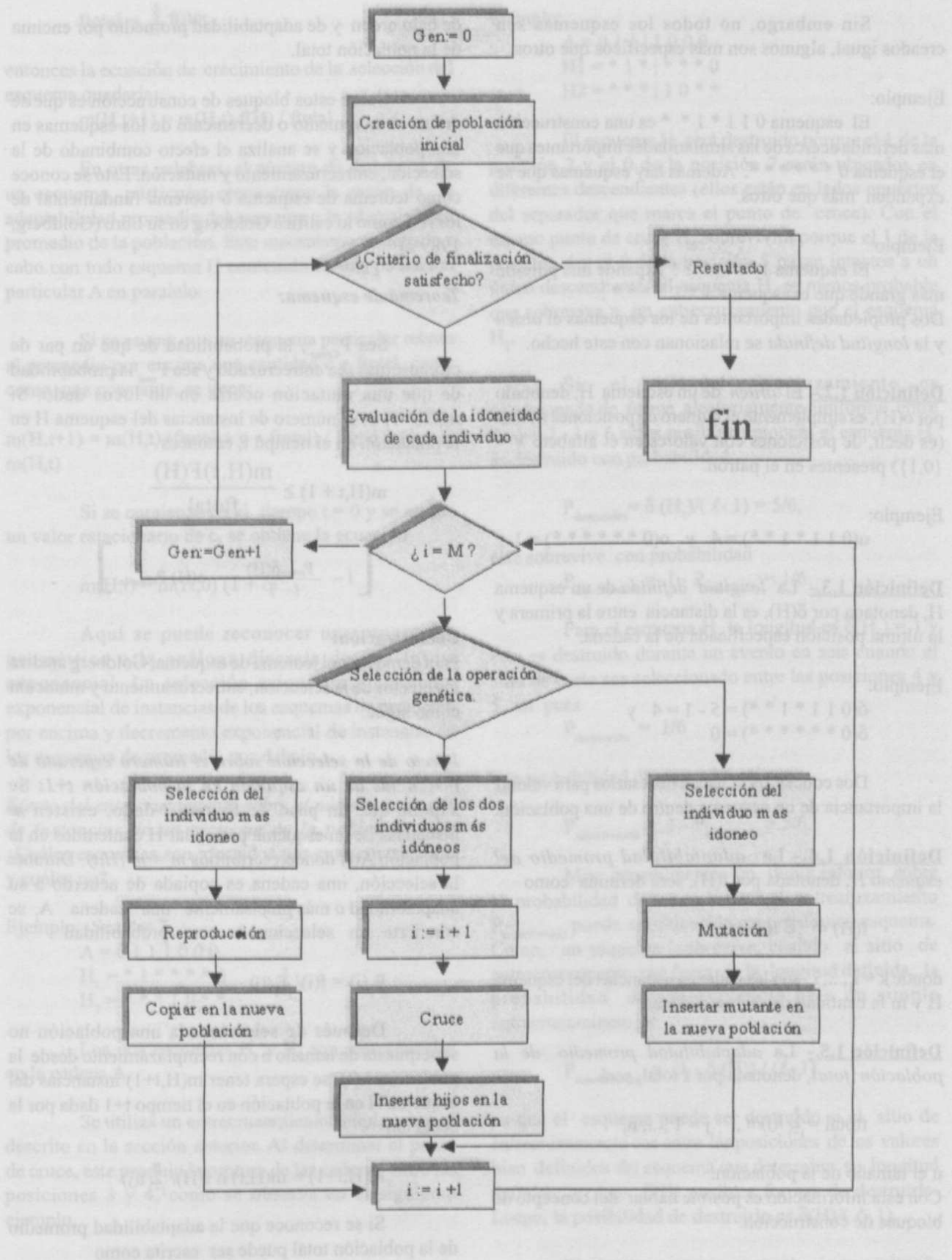
**Definición 1.1.-** Un *esquema* es una cadena de longitud  $\ell$  (la longitud de las cadenas de la población), tomada del alfabeto  $\{0,1,*\}$  donde  $*$  permite parear con 0 ó 1 en una posición particular. El  $*$  es un metasímbolo, que es una herramienta notacional para describir todas las posibles similitudes entre las cadenas.

Cada esquema representa el conjunto de todas las cadenas de longitud  $\ell$ , en cuyas posiciones con valores 0 y 1 son idénticas.

#### Ejemplo:

El esquema  $10^{**}1$ , representa el conjunto  $\{10001, 10011, 10101, 10111\}$ . Los elementos de este conjunto se conocen como instancias del esquema que ellos representan. Los esquemas son llamados también, *subconjuntos de similitud* porque ellos representan subconjuntos de cadenas con similitudes en ciertas posiciones fijadas. Las posiciones fijadas de un esquema son las posiciones de la cadena con valores 0 y 1.





Sin embargo, no todos los esquemas son creados igual, algunos son más específicos que otros.

Ejemplo:

El esquema 0 1 1 \* 1 \* \* es una construcción más definida acerca de las similitudes importantes que el esquema 0 \* \* \* \* \*. Además hay esquemas que se expanden más que otros.

Ejemplo:

El esquema 1 \* \* \* \* 1 \* expande una porción más grande que el esquema 1 \* 1 \* \* \* \*. Dos propiedades importantes de los esquemas el *orden* y la *longitud definida* se relacionan con este hecho.

**Definición 1.2.-** El *orden* de un esquema H, denotado por  $\alpha(H)$ , es simplemente el número de posiciones fijadas (es decir, de posiciones con valores en el alfabeto  $V = \{0,1\}$ ) presentes en el patrón.

Ejemplo:

$$\alpha(0 1 1 * 1 * *) = 4 \quad \text{y} \quad \alpha(0 * * * * *) = 1$$

**Definición 1.3.-** La *longitud definida* de un esquema H, denotada por  $\delta(H)$ , es la distancia entre la primera y la última posición especificada de la cadena.

Ejemplo:

$$\delta(0 1 1 * 1 * *) = 5 - 1 = 4 \quad \text{y} \\ \delta(0 * * * * *) = 0$$

Dos conceptos se hacen necesarios para valorar la importancia de un esquema dentro de una población.

**Definición 1.4.-** La *adaptabilidad promedio del esquema H*, denotada por  $f(H)$ , será definida como

$$f(H) = \frac{\sum_{k=1}^m f(k)}{m}$$

donde  $k = i_1, \dots, i_m$  son las cadenas instancias del esquema H y m la cantidad de dichas cadenas.

**Definición 1.5.-** La *adaptabilidad promedio de la población total*, denotada por  $f_{total}$ , será

$$f_{total} = \sum_{j=1}^n f(j)/n, \quad j = 1, \dots, n,$$

n el tamaño de la población.

Con esta información es posible hablar del concepto de bloques de construcción.

**Definición 1.6.-** Se denominan *bloques de construcción* a los esquemas que son de longitud definida pequeña,

de bajo orden y de adaptabilidad promedio por encima de la población total.

Sobre estos bloques de construcción es que se define el incremento o decremento de los esquemas en una población y se analiza el efecto combinado de la selección, entrecruzamiento y mutación. Esto se conoce como teorema de esquema o teorema fundamental de los AG como lo califica Goldberg en su libro (Goldberg, 1989 a)

**Teorema de esquema:**

Sea  $P_{Cruce}$ , la probabilidad de que un par de cromosomas sea entrecruzado y sea  $P_{mut}$  la probabilidad de que una mutación ocurra en un locus dado. Si  $m(H,t+1)$  es el número de instancias del esquema H en la población en el tiempo t, entonces

$$m(H,t+1) \geq \frac{m(H,t)F(H)}{f_{total}}$$

$$\left[ 1 - \frac{P_{Cruce}\delta(H)}{l-1} - \alpha(H)P_{mutación} \right]$$

**Demostración:**

Para demostrar el teorema de esquema, Goldberg analiza los efectos de la selección, entrecruzamiento y mutación como sigue:

**Efecto de la selección sobre el número esperado de instancias de un esquema en la población t+1:** Se supone que un paso de tiempo t dado, existen m instancias de un esquema particular H contenidos en la población A(t) donde escribimos  $m = m(H,t)$ . Durante la selección, una cadena es copiada de acuerdo a su adaptabilidad o más precisamente una cadena  $A_i$  se convierte en seleccionada con probabilidad

$$P_s(i) = f(i) / \sum_{j=1}^n f(j)$$

Después de seleccionada una población no superpuesta de tamaño n con reemplazamiento desde la población A(t), se espera tener  $m(H,t+1)$  instancias del esquema H en la población en el tiempo t+1 dada por la ecuación

$$m(H,t+1) = m(H,t) n f(H) / \sum_{j=1}^n f(j)$$

Si se reconoce que la adaptabilidad promedio de la población total puede ser escrita como

$$f_{total} = \sum_{j=1}^n f(j)/n$$

entonces la ecuación de crecimiento de la selección del esquema quedaría:

$$m(H,t+1) = m(H,t) f(H) / f_{total}$$

En otras palabras, el número de instancias de un esquema particular crece como la razón de la adaptabilidad promedio del esquema a la adaptabilidad promedio de la población. Este mecanismo es llevado a cabo con todo esquema H contenido en una población particular A en paralelo.

Si se asume que un esquema particular rebasa al promedio por encima una cantidad  $c \cdot f_{total}$ , con  $c$  como una constante, se tiene:

$$m(H,t+1) = m(H,t) (f_{total} + c \cdot f_{total}) / f_{total} = (1 + c) m(H,t)$$

Si se comienza en el tiempo  $t = 0$  y se asume un valor estacionario de  $c$ , se obtiene la ecuación

$$m(H,t) = m(H,0) (1 + c)^t$$

Aquí se puede reconocer una progresión geométrica o la análoga discreta de una forma exponencial. La selección asigna un incremento exponencial de instancias de los esquemas de promedio por encima y decremento exponencial de instancias de los esquemas de promedio por debajo.

*Efecto del entrecruzamiento sobre el número esperado de instancias de un esquema en la población  $t+1$ : ¿Cuáles esquemas son afectados por entrecruzamiento y cuáles no?*

Ejemplo: Sea  $\ell = 7$ ,

$$A = 0111000$$

$$H_1 = *1****0$$

$$H_2 = ****10**$$

Los dos esquemas  $H_1$  y  $H_2$  son representados en la cadena A.

Se utiliza un entrecruzamiento simple como el descrito en la sección anterior. Al determinar el punto de cruce, este produjo la ruptura de las cadenas entre las posiciones 3 y 4, como se observa en el siguiente ejemplo.

Ejemplo:

$$A = 0111|1000$$

$$H_1 = *1*|****0$$

$$H_2 = ****|10**$$

El esquema  $H_1$  será destruido porque el 1 de la posición 2 y el 0 de la posición 7 serán ubicados en diferentes descendientes (ellos están en lados opuestos del separador que marca el punto de cruce). Con el mismo punto de cruce  $H_2$  sobrevivirá porque el 1 de la posición 4 y el 0 de la posición 5 pasan intactos a un único descendiente. El esquema  $H_1$  es menos probable que sobreviva a un entrecruzamiento que el esquema  $H_2$ .

Si el sitio del entrecruzamiento es seleccionado de forma aleatoriamente uniforme entre  $\ell - 1 = 7 - 1 = 6$  sitios posibles, entonces el esquema  $H_1$  es destruido con probabilidad

$$P_{destrucción} = \delta(H_1) / (\ell - 1) = 5/6,$$

éste sobrevive con probabilidad

$$P_{supervivencia} = 1 - P_{destrucción} = 1/6.$$

Para el esquema  $H_2$  la longitud es  $\delta(H_2) = 1$  y éste es destruido durante un evento en seis cuando el sitio de corte sea seleccionado entre las posiciones 4 y 5, así pues

$$P_{destrucción} = 1/6$$

o la probabilidad de supervivencia es

$$P_{supervivencia} = 1 - P_{destrucción} = 5/6.$$

Más generalmente, un límite inferior sobre la probabilidad de supervivencia al entrecruzamiento  $P_{supervivencia}$  puede ser calculada para cualquier esquema. Como, un esquema sobrevive, cuando el sitio de entrecruzamiento cae fuera de la longitud definida, la probabilidad de supervivencia bajo un simple entrecruzamiento es:

$$P_{supervivencia} = (1 - \delta(H)) / (\ell - 1)$$

ya que el esquema puede ser destruido si el sitio de entrecruzamiento cae entre las posiciones de los valores bien definidos del esquema que determina su longitud definida y hay  $\delta(H)$  posibles formas de destruirlo. Luego, la posibilidad de destruirlo es  $\delta(H) / (\ell - 1)$ .

Si el entrecruzamiento es el mismo realizado por selección aleatoria digamos con probabilidad  $P_{Cruce}$



en un pareo particular, la probabilidad de supervivencia puede darse por la expresión

$$P_{\text{supervivencia}} > 1 - P_{\text{Cruce}} * \delta(H) / (\ell - 1)$$

la cual se reduce a la primera expresión cuando  $P_{\text{Cruce}} = 1.0$ .

El efecto combinado de la selección y entrecruzamiento puede considerarse en la siguiente expresión:

$$m(H, t + 1) \geq \frac{m(H, t) F(H)}{f_{\text{total}}} \left[ 1 - \frac{P_{\text{Cruce}} \delta(H)}{\ell - 1} \right]$$

El esquema H crece o muere dependiendo de un factor de multiplicación. Ahora, este factor depende de si el esquema está por encima o por debajo del promedio de la población y si el esquema tiene longitud definida relativamente pequeña o grande. Claramente, aquellos esquemas con promedio por encima y longitud definida pequeña, serán muestreados con una razón de incremento exponencial.

*Efecto de la mutación sobre el número de instancias de un esquema en la población t+1:* Con el objetivo de que un esquema H sobreviva, todas las posiciones fijadas necesitan ellas mismas sobrevivir. Por tanto desde que un simple alelo sobreviva con probabilidad  $(1 - P_m)$  y desde que cada una de las mutaciones es estadísticamente independiente, un esquema particular sobrevivirá cuando cada una de las  $o(H)$  posiciones fijadas en el esquema sobrevivan. Multiplicando la probabilidad de supervivencia  $(1 - P_m)$ ,  $o(H)$  veces obtenemos la probabilidad de supervivencia de la mutación,  $(1 - P_m)^{o(H)}$ . Para valores pequeños de  $P_m$  ( $P_m \ll 1$ ), la probabilidad de supervivencia del esquema puede aproximarse por la expresión  $1 - o(H) P_m$ .

Por tanto se llega a que un esquema particular H recibe un número esperado de instancias en la próxima generación bajo selección, entrecruzamiento y mutación, dada por la siguiente ecuación (ignorando pequeños términos del producto cruzado):

$$m(H, t + 1) \geq \frac{m(H, t) F(H)}{f_{\text{total}}} \left[ 1 - \frac{P_{\text{Cruce}} \delta(H)}{\ell - 1} - o(H) P_{\text{mutación}} \right]$$

Las conclusiones son las siguientes: Los esquemas cortos, de bajo orden, de promedio por encima, reciben un incremento exponencial de instancias

en las subsecuentes generaciones y contrariamente los esquemas con promedio por debajo reciben un decremento exponencial de instancias en el tiempo. En (Holland, 1992) se asocia la competencia de los esquemas en los AG al problema del bandido multi-brazos de la teoría de la decisión estadística.

En el trabajo (Goldberg, 1989a) se expone la hipótesis de los bloques de construcción, la cual establece que, «los bloques de construcción se combinan para formar mejores cadenas». Esto significa que la recombinación y el crecimiento exponencial de los bloques de construcción conduce a la formación de mejores bloques de construcción, los cuales aseguran soluciones finales altamente adaptables. En el propio trabajo Goldberg simula el procesamiento de los esquemas en un AG simple. Un estimado de la cantidad de bloques de construcción procesados por el AG se reporta en (Holland, 1992) sobre el orden de  $n^3$  esquemas. Holland identifica esta propiedad como el *paralelismo implícito*.

Finalmente, el teorema de esquema predice el crecimiento de los esquemas, sin embargo tiene limitaciones a la hora de describir el funcionamiento del AG. Sus principales inconvenientes son los siguientes:

- 1.- Tanto  $f(H)$  como  $f_{\text{total}}$  no permanecen constantes de generación en generación. Por ejemplo, las adaptabilidades de los individuos de la población cambian significativamente después de las primeras generaciones.
- 2.- El teorema de esquema explica el esquema perdido y no el esquema ganado, ya que los esquemas son construidos por entrecruzamiento y mutación. Además, cuando el AG avanza en su ejecución los individuos en la población se hacen cada vez más parecidos entre sí, y los esquemas que se destruyen por el entrecruzamiento pronto son recuperados.
- 3.- El teorema de esquema es una teoría del valor esperado y no tiene en cuenta la varianza. En muchos problemas cuando la varianza de la adaptabilidad del esquema es alta, la detección de los esquemas que contienen el óptimo global se convierte en un proceso ruidoso (Goldberg y Rudnick, 1991). La varianza de la adaptabilidad del esquema puede conducir a una convergencia subóptima, es decir, la conocida convergencia prematura.

Sin embargo, el teorema de esquema describe importantes aspectos del funcionamiento del AG. Estos pueden ser resumidos en los siguientes puntos:

- 1.- Se puede establecer que las probabilidades de mutación más altas incrementan la destrucción de los esquemas de un orden mayor, por otro lado, las probabilidades de entrecruzamiento más altas incrementan la destrucción de esquemas de una longitud definida mayor.
- 2.- Cuando se realiza la selección la población converge en una razón proporcional a la proporción entre la adaptabilidad del mejor individuo y la adaptabilidad promedio de la población. Esto es una medida de la carga o peso de la selección (Back, 1994).
- 3.- Incrementando la probabilidad de entrecruzamiento o la probabilidad de mutación o decrementando la carga de la selección se llega a incrementar el muestreo o exploración del espacio de búsqueda, pero no se permite una mayor explotación de los buenos esquemas que el AG localiza.
- 4.- Decrementando la probabilidad de entrecruzamiento o la probabilidad de mutación o incrementando la carga de la selección, se llega a un uso incrementado o explotación de los mejores esquemas, pero no se permite una mayor exploración del espacio de búsqueda. Estos dos últimos puntos expresan el delicado problema del balance entre la exploración y explotación.

### 1.5 AG en problemas de secuenciación

En este apartado se analizan las características de los AG diseñados para los problemas de secuenciación. Se entienden como problemas de secuenciación u ordenamiento, aquellos en los cuales las soluciones buscadas son secuencias ordenadas de objetos tal que la ordenación dada cumple con alguna medida de optimalidad. En ellos se involucran las permutaciones. Como ejemplos podemos citar, el problema del viajante de comercio (PVC), el problema de coloreo de un grafo, el empaqueo de objetos o "bin-packing", el problema de programación de tareas y el problema de búsqueda de un ciclo hamiltoniano, entre otros. Todos ellos clasificados en la clase de problemas NP-completos (Garay y Johnson, 1979). La aplicación de los AG a problemas de secuenciación ha tropezado con la dificultad de que su representación no se ajusta fácilmente a los AG. El problema radica en el hecho de que la utilización del entrecruzamiento tradicional produce soluciones ilegales que no contienen a todos los elementos de la permutación y contienen elementos duplicados (Goldberg, 1989a). En esta clase de problemas un cromosoma es una secuencia ordenada de objetos, en este sentido lo tradicional es poner etiquetas numéricas a dichos objetos.

Motivados por los principios en que se basa el operador de entrecruzamiento tradicional de incorporar los mejores atributos de dos padres en un nuevo individuo, las primeras investigaciones se orientaron a crear operadores análogos que produjeran hijos legales. Esto condujo a la proliferación de diferentes formas de representar un cromosoma.

En el presente análisis se parte de un conjunto de diferentes representaciones del cromosoma para los problemas de secuenciación reportado en (Michalewicz, 1992). En su libro Michalewicz define las 5 primeras categorías aquí expuestas. A las mismas añadiremos nuevos operadores encontrados en la literatura ubicándolos en la representación apropiada. Cinco nuevas categorías son añadidas con sus respectivos operadores. Se utilizan diferentes criterios de agrupación de las categorías aquí expuestas, destacando las principales tendencias de trabajo.

### CONCLUSIONES

La idea principal que podemos tomar de esta reseña sobre algoritmos genéticos, es que, aún siendo una técnica relativamente nueva no sólo en México sino en el mundo entero, ha cobrado una gran popularidad por las grandes ventajas que nos ofrecen este tipo de técnicas en cuanto a problemas NP-completos, es decir para la resolución de problemas donde la solución es permutacional y el recorrido posible para encontrar la solución óptima es demasiado grande.

Actualmente en nuestra Universidad se empiezan a crear los primeros grupos de trabajo de investigación relacionados con este tipo de técnicas y en poco tiempo creemos que empezaremos a obtener muy buenos resultados hacia problemas reales con los diferentes sectores de la localidad.

Y en un futuro (dos a cuatro años), tenemos planeado ser investigadores activos en esta líneas de investigación, donde existe mucho por hacer.

### BIBLIOGRAFÍA

- Antonoisse, J. (1989): A new interpretation of schema notation that overturns the binary encoding constraint. *Proc. of 3er Intl. Conference on Genetic algorithms*. J. D. Schaffer (Ed.). Morgan Kaufmann, Los Altos. 70-79.
- Back, T. (1992): The interaction of mutation rate, selection, and self-adaptation within a genetic algorithm. R. Manner y B. Manderick (Eds.), *Parallel problem solving from nature, 2*. Amsterdam: Elsevier. 85-94



- Back, T. (1993): Optimal mutation rates in genetic search. *Proc. 5th Intl. Conf. on Genetic Algorithms and their Applications*. Morgan Kaufmann, San Mateo. 2-8.
- Back, T. (1994): Selective pressure in evolutionary algorithms: A characterization of selection mechanisms. *Proceeding of the First IEEE Conference on Evolutionary Computation*, IEEE World Congress on Computational Intelligence, 57-62.
- Baker, J. E. (1985): Adaptive selection methods for genetic algorithms. *Proc. of an International Conference on Genetic Algorithms and their applications*. J.J. Grefenstette (Ed). Lawrence Erlbaum Associates, Hillsdale, NJ.
- Bautista, E. y Valenzuela, M. (1992): Optimización de permutaciones con algoritmos genéticos: problemas de programación de tareas. *Memorias III Congreso Iberoamericano de Inteligencia Artificial*. Editorial Limusa, S. A. de C. V. Megabyte, Grupo Noriega Editores. 217-229.
- Booker, L.B. (1987): Improving search in genetic algorithms. *Genetic algorithms and Simulated Annealing*. L. Davis (Ed.). Morgan Kaufmann, Los Altos, CA. 61-73.
- Bruck, J.; Cypher, R. y Ho, Ch. T. (1995): On the construction of fault-tolerant cube-connected cycles networks. *Journal of parallel and distributed computing* 25. 98-106.
- Bruck, J.; Cypher, R. y Soroker, D. (1992): Tolerating faults in hypercubes using subcubes partitioning. *IEEE Transaction on computers*, Vol.41, No. 5.
- Bukatova, I. L. y Gulyaev, Yu. V. (1993): From genetic algorithms to evolutionary computer. *Proc. 5th Intl. Conf. on Genetic Algorithms and their Applications*. Morgan Kaufmann, San Mateo. 613-617.
- Carruana, R.A. y Schaffer, J.D. (1988): Representation and hidden bias: Gray vs. binary coding for genetic algorithms. *Proc. 5th Intl. Conf. on Machine Learning*. Morgan Kaufmann, Los Altos, CA.
- Celikis, G. y Kuehni, R.G. (1983): Color Technology in the Textile Industry, US.
- Davis, L. (1985): Applying adaptive algorithms to epistatic domains. *Proc. Intl. Joint Conf. On Artificial Intelligence*. 162-164.
- Davis, L. (1991): Handbook of genetic algorithms. New York: Van Nostrand Reinhold.
- De Jong, K. y Spears, W. (1989): Using genetic algorithms to solve NP-complete problems. *Proc. of 3er Intl. Conference on Genetic algorithms*. J. D. Schaffer (Ed.). Morgan Kaufmann, Los Altos. 124-132.
- De Jong, K. y Spears, W. (1990): An analysis of the interacting roles of population size and crossover in genetic algorithms. *Proc. First Workshop Parallel Problem Solving from Nature*, Springer-Verlag, Berlin. 38-47.
- Eshelman, L. J. y Schaffer, J.D. (1993): Real-coded genetic algorithms and interval-schemata. In L. D. Whitley (Ed.), *Foundations of genetic algorithms, 2*. San Mateo: Morgan Kaufmann. 187-202.
- Forrest, S. (1993): Genetic Algorithms: Principles of natural selection applied to computation. *Science*, Vol. 261, August. 872-878.
- Fox, B. R. y McMahon, M. R. (1991): Genetic operators for sequencing problems. G.J.E. Rawlins (Ed.) *Foundations of genetic algorithms*. San Mateo: Morgan Kaufmann. 284-300.
- Garay, M.R. y Johnson, D.S. (1979): Computers and intractability. A guide to the theory of NP-Completeness. W. H. Freeman and Company, New York.
- Goldberg, D. E. (1989a): Genetic algorithms in search, optimization, and machine learning. Reading, MA: Addison-Wesley.
- Goldberg, D. E. (1989b): Sizing populations for serial and parallel genetic algorithms. *Proc. of 3er Intl. Conference on Genetic algorithms*. J. D. Schaffer (Ed.). Morgan Kaufmann, Los Altos, CA. 70-79.
- Goldberg, D. E. (1989c): Genetic algorithms and Walsh functions: Part I, a gentle introduction. *Complex Systems*, 3, 129-152.
- Grefenstette, J.J. (1986): Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man and Cybernetics*, 16(1), 122-128.
- Hável, I. (1982): Semipaths in directed cubes, en *Graphs and other combinatorial topic*, B. 59 (Teubner Texte zum Mathematik, Teubner, Leipzig) (3º Symposium Tchescoslovaque de Th. Graphes, Nov. 1982).
- Holland, J. H. (1975): Adaptation in natural and artificial systems. Ann Arbor: University of Michigan Press.
- Holland, J. H. (1992): Adaptation in natural and artificial systems. Cambridge, MA: MIT Press.
- Koza, J. R. (1992): Genetic programming: On the programming of computer by means of natural selection. Cambridge, MA: MIT Press.
- Michalewicz, Z. (1992): Genetic algorithms + data structures = evolutionary programs, Springer-Verlag, Berlin.
- Morales M., L.; Maldonado E., F.; Radillo R., R. y Ciurlizza G., A. (en prensa): Optimization of color sequence in the process of fabric dyeing, 1997.
- Mühlenbein, H.; Gorges-Schleuter, M. y Krmer, O. (1988): Evolution algorithms in combinatorial optimization, *Parallel Computing*, Vol. 7. 65-85.
- Ochoa, A. (1997): Partial evaluation and evolutionary computation. Fourth International conference on Artificial Life. 27-30 julio, Londres.
- Ordoñez, I. y Valenzuela, M. (1992): Optimización de permutaciones con algoritmos genéticos: problemas de vendedor viajero. *Memorias III Congreso Iberoamericano de Inteligencia Artificial*. Editorial Limusa, S. A. de C. V. Megabyte, Grupo Noriega Editores. 271-282.
- Padilla, F. (1998): Los algoritmos Genéticos en México. Congreso Internacional de Inteligencia Artificial, de la Habana, Cuba.
- Ponce de León, E. (1997): La Conjetura de Erdős y Hável: acciones de grupo, representación circular y grafos multi-niveles. *Reporte de Investigación. ICIMAF 97-39, CENIA 97-02*. ISSN 0138-8916. Marzo.
- Ponce de León, E.; Ochoa, A. y Santana, R. (1995): A genetic algorithm for a Hamiltonian path problem. *Reporte de Investigación. ICIMAF 95-01, CENIA 95-01*. ISSN 0138-8916. Septiembre.
- Poon, P. W. y Carter, J. N. (1995): Genetic algorithm crossover operators for ordering applications. *Computers Ops. Res.* Vol. 22, No. 1. 135-147.
- Santana, R. y Ponce de León, E. (1996): Algoritmos genéticos para un problema de búsqueda de caminos Hamiltonianos. *Tesis de diploma. Facultad de Matemática*. Universidad de la Habana. Mayo.
- Santana, R. y Ponce de León, E. (1997): An evolutionary optimization approach for detecting structures on graphs. *Reporte de Investigación ICIMAF*. En proceso de revisión.
- Schaffer, J.D.; Caruana, R.A.; Eshelman, L.J. y Das, R. (1989): A study of control parameters affecting online performance of genetic algorithms for function optimization. *Proc. of 3er Intl. Conference on Genetic Algorithms*. J. D. Schaffer (Ed.). Morgan Kaufmann, Los Altos, CA. 51-60.
- Shaefer, C.G. (1987): Proc of 2nd Intl. Conference on Genetic Algorithms. J.J. Grefenstette(Ed.). Erlbaum, Hillsdale, NJ. 50-58.