

# ALGORITMOS GENÉTICOS: QUÉ SON Y CÓMO FUNCIONAN

NESTOR MANUEL GARZÓN TORRES<sup>1</sup>  
LUIS CARLOS TORRES SOLER<sup>2</sup>

## Resumen

*El ensayo siguiente muestra cómo los algoritmos genéticos (GA) toman un papel muy importante en los nuevos sistemas de información. También se muestra la estructura básica de un algoritmo genético y su aplicación en las soluciones de problemas computacionales.*

## Abstract

*The following essay shows how the genetic algorithms (GA) took a very important role in the new information systems. Also we shows the basic structure of an genetic algorithm and their application in the solutions of computational problems.*

## INTRODUCCIÓN

Los Algoritmos Genéticos están inspirados en el principio de la evolución de las especies planteado por Darwin y en la genética. Son algoritmos probabilísticos que ofrecen un mecanismo de búsqueda paralela y adaptativa basada en el principio de supervivencia de los más aptos y en la reproducción.

## Historia

Hoy día existe un amplio y creciente interés por la utilización de los algoritmos genéticos, es una nueva técnica de búsqueda basada en la teoría de la evolución. Esta técnica emplea los mecanismos de selección que utiliza la naturaleza, los individuos más aptos de una población son los que sobreviven al adaptarse más fácilmente a los cambios que se producen en su entorno. Se conoce que estos cambios se efectúan en los genes (unidad básica de codificación de cada uno de los atributos de un ser vivo) de un individuo, y que los atributos más deseables (los que le permiten a un individuo adaptarse mejor a su entorno) del mismo se transmiten a sus descendientes, cuando éste se reproduce. John Holland, investigador de la Universidad de Michigan, consciente de la importancia de la selección natural, a finales de 1960 desarrolló la técnica llamada “planes reproductivos” incorporándola en un programa de computador. Su objetivo era que los computadores aprendieran por sí mismos. Esta técnica se hizo popular bajo el nombre de algoritmos genéticos desde 1975.

## Definición

Una definición bastante completa de un algoritmo genético es: *un proceso matemático altamente paralelo que transforma un*

*conjunto de objetos individuales con respecto al tiempo usando operaciones modeladas de acuerdo al principio Darwiniano de reproducción y supervivencia del más apto, y tras haberse presentado de forma natural una serie de operaciones genéticas. Cada uno de estos objetos suele ser una cadena de caracteres (letras o números) de longitud fija que se ajusta al modelo de las cadenas de cromosomas, y se les asocia con una cierta función matemática que refleja su aptitud.*

Los principios de la naturaleza en los cuales están inspirados los algoritmos genéticos son muy simples: de acuerdo con la teoría de Darwin, el principio de selección privilegia los individuos más aptos con mayor longevidad y, por lo tanto, con mayor probabilidad de reproducción. Los individuos seleccionados para reproducirse tienen más chances de transmitir sus códigos genéticos a las próximas generaciones. Tales códigos genéticos constituyen la identidad de cada individuo y están representados en los cromosomas.

Estos principios se imitan en la construcción de algoritmos computacionales que buscan la mejor solución para un determinado problema, a través de la evolución de poblaciones de soluciones codificadas en cromosomas.

<sup>1</sup> Ingeniero de sistemas, especialista en redes de comunicación, estudios de doctorado en Sistemas de información, Profesor universitario, Investigador. nestormg@gmail.com

<sup>2</sup> Matemático, MSc. Ingeniería de sistemas, MA. Ciencias de la educación, Profesor universitario, Investigador. lctorress@gmail.com

En los algoritmos genéticos, cada cromosoma es una estructura de datos que representa una de las posibles soluciones del problema. Los cromosomas se someten a un proceso de evolución que envuelve evaluación, selección, recombinación sexual (crossover) y mutación. Después de varios ciclos de evolución la población deberá contener individuos más aptos.

### Elementos de un algoritmo genético

Se caracterizan los Algoritmos Genéticos a través de los siguientes componentes:

1. Problema a ser solucionado.
2. Representación de soluciones del problema.
3. Decodificación del cromosoma.
4. Evaluación.
5. Selección.
6. Operadores genéticos.
7. Inicialización de la población.
8. Parámetros y criterios de parada.

#### 1. Problema

Los Algoritmos genéticos son particularmente aplicados en problemas complejos de optimización: problemas con diversos parámetros o características que precisan ser combinadas en busca de la mejor solución; problemas con muchas restricciones o condiciones que no pueden ser representadas matemáticamente y problemas con grandes espacios de búsqueda.

Los Algoritmos genéticos han sido aplicados en diversos problemas de optimización, tales como: funciones matemáticas, combinatoria, planeamiento, problema del cajero viajero, rutas de vehículos, layout de circuitos, distribución, negocios, síntesis de circuitos electrónicos, etc.

#### 2. Representación

La representación de las posibles soluciones dentro del espacio de búsqueda de un problema define la estructura del cromosoma que va ser manipulado por el algoritmo. Normalmente, la representación binaria es la más empleada por ser más simple, fácil de manipular a través de los operadores genéticos, fácil de ser transformada en entero o real y además, por facilitar la demostración de los teoremas. Por ejemplo, sea el problema de encontrar el valor máximo de la función  $f(x) = x^2$ , siendo  $x$  entero en  $[0,63]$ . Se pueden representar las soluciones del problema a través de un cromosoma de 6 bits.

C1                    0 0 1 0 0 1                     $x=9$   
 C2                    0 0 0 1 0 0                     $x=4$

Un número binario también puede representar un número real  $X \in [X_{min}, X_{max}]$ , con precisión de  $p$  cifras decimales. Para eso son necesarios  $K$  bits, calculados por la siguiente relación:

$$2^k \leq (X_{max} - X_{min}) 10^p$$

La representación binaria, sin embargo, no puede ser empleada siempre; muchas veces el problema exige un alfabeto de representaciones con más símbolos. Cualquiera que sea la representación escogida, debe ser capaz de representar todo el espacio de búsqueda que se desea investigar.

### 3. Decodificación

La decodificación del cromosoma consiste básicamente en la construcción de la solución real del problema. El proceso de decodificación construye la solución para que ésta sea evaluada por el problema. La ventaja de la representación binaria es la gran facilidad para ser transformada en entero o en real.

En la transformación en un número real, se considera el intervalo de valores o longitud del dominio ( $C$ ) de los reales de tal forma que

$$X_k = X_b \frac{C}{2^n - 1} - X_{min}$$

Donde  $X_k \in [X_{min}, X_{max}]$ ;  $X_b$  es el entero correspondiente al binario;  $n$  es el número de bits del cromosoma; y  $C$  es la amplitud del dominio de la variable  $X$ , dado por  $C = [X_{max} - X_{min}]$ .

### 4. Evaluación

La evaluación es la unión entre el algoritmo genético y el mundo externo. La evaluación se realiza a través de una función que representa de forma adecuada el problema y tiene como objetivo suministrar una medida de aptitud de cada individuo en la población actual. La función de evaluación es para un algoritmo genético lo que el medio ambiente es para los seres humanos. Las funciones de evaluación son específicas de cada problema. En el ejemplo, la función matemática  $f(x)=x^2$  mide la aptitud de cada individuo. C1 es un individuo más apto que C2.

	Cromosoma							x	f(x)
C1	0	0	0	1	0	0	1	9	81
C2	0	0	0	0	1	0	0	4	16

### 5. Selección

El proceso de selección en los algoritmos genéticos selecciona individuos para la reproducción. La selección está basada en la aptitud de los individuos: individuos más aptos tienen mayor probabilidad de ser escogidos para la reproducción.

Así, si  $f_i$  fuera la evaluación del individuo  $i$  en la población actual, la probabilidad  $p_i$  de ser seleccionado el individuo  $i$  sería proporcional a:

$$p_i = f_i / \sum f_i$$



Donde  $n$  es el número de individuos en la Población. La selección en los Algoritmos genéticos se implementa normalmente por una ruleta donde cada individuo es representado por una porción proporcional a su evaluación relativa.

### 6. Operadores genéticos

Los diferentes métodos u operaciones que se pueden ejercer sobre una población y que permite obtener poblaciones nuevas se denominan operadores genéticos. Una vez que se ha evaluado cada individuo con una función de aptitud (*fitness*), se aplican estos operadores. Se destacan los siguientes operadores.

**Operador de Selección.** Es el que se utiliza en el proceso de selección.

**Operador de Cruce.** Consiste en unir en alguna forma los cromosomas de los padres que han sido previamente seleccionados para formar dos descendientes. Existen diversas variaciones, dependiendo del número de puntos de división a emplear y la forma de ver el cromosoma. El operador cruce se aplica en dos pasos: en el primero los individuos se aparean (se seleccionan de dos en dos) aleatoriamente con una determinada probabilidad, llamada probabilidad de cruce  $p_c$ ; en el segundo paso a cada par de individuos seleccionados anteriormente se le aplica un intercambio en su contenido desde una posición aleatoria  $K$  hasta el final, con  $K \in [1, m - 1]$ , donde  $m$  es la longitud de individuo,  $K$  es el denominado punto de cruce y determina la subdivisión de cada padre en dos partes que se intercambian para formar dos nuevos hijos, tal y como lo podemos ver en la figura 1. Esto se conoce como cruce ordinario o cruce de un punto. El objetivo del operador de cruce es recombinar subcadenas de forma eficiente; esta gestión recibe el nombre de construcción de bloques.

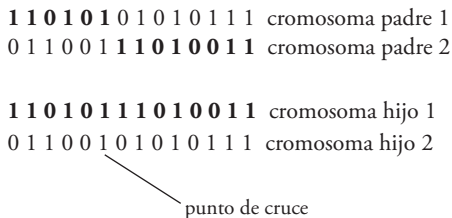


Figura 1. Cruce de cromosomas.

**Operador de Mutación.** Consiste en la alteración aleatoria de cada uno de los genes del individuo con una probabilidad de mutación  $p_m$ . El objetivo de la mutación es producir diversidad en la población. Si al generar aleatoriamente la población inicial o después de varias generaciones, en la misma posición de todos los cromosomas sólo aparece un único elemento del alfabeto utilizado, esto supondrá que con los operadores de reproducción y cruce nunca cambiara dicho elemento, por lo que puede ocurrir que jamás se alcance la solución más optima a nuestro problema. La probabilidad de aparición del operador

de mutación no debe ser grande para no perjudicar la correcta construcción de bloques. El operador de mutación origina variaciones elementales en la población y garantiza que cualquier punto del espacio de búsqueda pueda ser alcanzado.

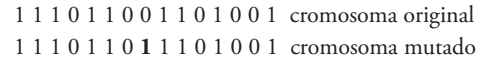


Figura 2. Mutación de cromosomas.

### 7. Inicialización de la Población

La inicialización de la población determina el proceso de creación de los individuos para el primer ciclo del algoritmo. Normalmente, la población inicial se forma a partir de individuos creados aleatoriamente. Las poblaciones iniciales creadas aleatoriamente pueden ser sembradas con cromosomas buenos para conseguir una evolución más rápida, si se conocen *a priori*, el valor de estas “semillas” buenas.

### 8. Parámetros y Criterios de parada

En un algoritmo genético varios parámetros controlan el proceso de evolución:

- Tamaño de la población. Número de puntos del espacio de búsqueda siendo considerados en paralelo.
- Tasa de cruce. Probabilidad de un individuo de ser recombinado con otro.
- Tasa de mutación. Probabilidad de que el contenido de cada posición/gen del cromosoma sea alterado.
- Número de generaciones. Número total de ciclos de evolución de un algoritmo genético.
- Total de individuos. Número total de tentativas ([tamaño de la población] x [número de generaciones]).

Los dos últimos parámetros son empleados generalmente como criterio de parada de un algoritmo genético. Un algoritmo genético puede ser descrito como un proceso continuo que repite ciclos de evolución controlados por un criterio de parada, como se muestra en la siguiente 3.

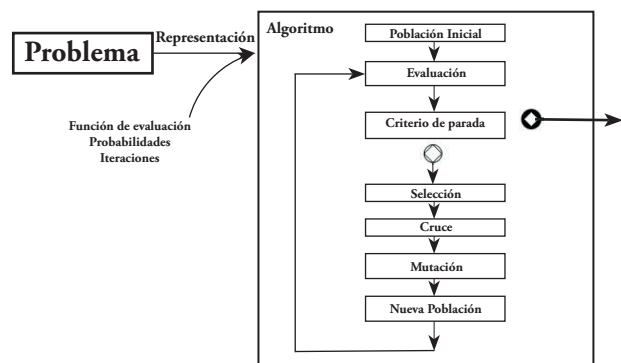


Figura 3. Proceso de los Algoritmos Genéticos.

## Desempeño de los algoritmos genéticos

Los algoritmos genéticos son sistemas no lineales con comportamiento fuertemente biológico. Los Algoritmos genéticos combinan cambios aleatorios con procesos probabilísticos. Los Algoritmos genéticos son, por lo tanto, estocásticos: difícilmente repiten un resultado de un experimento.

El desempeño de un algoritmo genético es medido por el grado de evolución alcanzado durante todo el proceso de evolución (experimento). Dada la naturaleza estocástica de los Algoritmos genéticos es necesario evaluar el resultado medio de varios experimentos de un algoritmo genético, para tener una idea de su desempeño.

Las principales medidas de desempeño son:

1. Curva de la media de los mejores cromosomas en cada ciclo durante varios experimentos. Presenta el desempeño medio de un algoritmo genético y sirve para el ajuste de los parámetros.
2. Curva *on-line* de la media de la evaluación de todos los individuos hasta un determinado instante en un experimento. Mide la velocidad con que el algoritmo genético consigue producir buenas soluciones para el uso "*on-line*" de las soluciones.
3. Curva *off-line* de la media de la evaluación de los mejores individuos hasta la generación actual, en un experimento. Mide el grado de convergencia del algoritmo genético en la creación de soluciones más aptas, generadas "*off-line*" con relación al problema.

## Técnicas y parámetros genéticos

Las técnicas, los parámetros y los tipos de operadores genéticos afectan significativamente el desempeño de un algoritmo genético. En general, el algoritmo genético es extremadamente sensible a la introducción o combinación de las técnicas empleadas. La selección de las técnicas, los parámetros y los tipos de operadores es empírica, sin embargo, está relacionada con el tipo de problema.

Podemos clasificar las técnicas empleadas en los algoritmos genéticos en las siguientes clases:

- 1- Técnicas de reproducción.
- 2- Técnicas de aptitud.
- 3- Técnicas de interpolación de parámetros.

## Técnicas de reproducción

Estas técnicas determinan el criterio de sustitución de los individuos de una población para la próxima generación. Existen básicamente los siguientes métodos:

- 1) *Cambio de toda población*: En cada ciclo, son creados N nuevos individuos sustituyendo la población anterior: se escogen N/2 pares para el apareamiento, generando N descendientes.

2) *Cambio de toda la población con elitismo*: Todos los cromosomas son sustituidos, copiándose directamente el cromosoma más apto de la población actual en la población siguiente.

3) *Cambio parcial de la población*: Genera M individuos ( $M < N$ ), que sustituyen los peores individuos de la población actual. Técnica elitista que mantiene la población más estática, permitiendo, por lo tanto, la utilización de operadores menos conservadores como el cruce uniforme.

4) *Cambio parcial de la población sin duplicados*: No se permite en la población la presencia de individuos duplicados. Se garantiza de este modo el mejor aprovechamiento del paralelismo intrínseco de los Algoritmos genéticos (N puntos diferentes del espacio de búsqueda siendo evaluados en cada ciclo). Sin embargo, implica un "*overhead*" computacional para la detección de los individuos duplicados.

## Técnicas de aptitud

Se trata de la manera por la cual son atribuidas numéricamente las aptitudes de los cromosomas en una población. Intuitivamente, el método más simple de atribuir una aptitud a un cromosoma es utilizar el valor numérico del resultado de la evaluación de este cromosoma por la función de evaluación. Existen, sin embargo, dos problemas importantes asociados a este método:

■ *Competición próxima*: individuos cuya aptitud relativa son próximas numéricamente.

■ *Súper-individuos*: individuos con evaluación muy superior a la media, capaces de dominar el proceso de selección, haciendo que el algoritmo genético converja prematuramente hacia un óptimo local.

Para resolver estos problemas se utilizan métodos de transformación de la evaluación numérica de los cromosomas en una aptitud. Los métodos más empleados son: "*windowing*", "*rankbased*" y la normalización lineal. Entre estos, la normalización lineal es el método más simple, eficiente para los tres problemas y es parametrizable.

El método de normalización lineal consiste en ordenar los cromosomas en orden creciente de evaluación ( $i=0$  corresponde al individuo menos apto) y atribuir valores de aptitud a los cromosomas linealmente entre un valor mínimo y un máximo, distanciados un valor fijo (tasa de decremento).

$$A_i = \min + \frac{\max - \min}{N - 1} X (i - 1)$$

Este método reduce el dominio ejercido por los súper-individuos y aumenta la presión selectiva entre los individuos con evaluación próxima en función de la tasa de decremento.

## Técnicas de interpolación de los parámetros

La interpolación de los parámetros de un algoritmo genético tiene como objetivo buscar el valor ideal de un determinado parámetro para cada ciclo, durante toda la evolución. Durante la evolución de un algoritmo genético algunos aspectos están altamente relacionados: La convergencia del algoritmo, la aptitud de los cromosomas, las tasas y parámetros. Por ejemplo, es intuitivamente perceptible que la tasa de aplicación del cruce sea mayor en las primeras generaciones mientras la población se presenta dispersa en el espacio de búsqueda. Después de varias generaciones los individuos tienden a presentar, en su mayoría, características muy similares. Un aumento de la tasa de mutación en esta fase de la evolución servirá para dispersar la población, trayendo nuevo material genético para la formación de mejores individuos. La interpolación de los parámetros puede ser lineal o adaptativa. En la interpolación lineal la tasa o parámetro es variado linealmente entre un valor inicial y final, a través de ajustes fijos, a cada  $K$  generaciones. La interpolación adaptativa, normalmente es empleada para el ajuste de la tasa de aplicación de los operadores, considerando el desempeño de estos operadores en los ciclos anteriores. Este desempeño es medido en función del éxito de estos operadores en la creación de mejores individuos. Este esquema requiere un proceso de puntuación de los individuos y sus descendientes, necesiéndose un razonable “*overhead*” computacional. El objetivo, de todo esto es alcanzar trayectorias óptimas para cada parámetro, durante toda la evolución.

## Clasificación

Los algoritmos genéticos según el proceso de selección, de cruce, de reemplazo u otros procesos se denominan de distinta forma.

- **Generacionales.** Se asemejan a la forma de reproducción de los insectos, donde una generación pone huevos, se aleja geográficamente o muere y es sustituida por una nueva. En este momento se realizan cruces en una piscina de individuos, los descendientes son puestos en otra, al final de la fase reproductiva se elimina la generación anterior y se pasa a utilizar la nueva.
- **De Estado Fijo.** Utilizan el esquema generacional de los mamíferos y otros animales de vida larga, donde coexisten padres y sus descendientes, permitiendo que los hijos sean educados por sus progenitores, pero también que a la larga se genere competencia entre ellos.

En este modelo, no solo se deben seleccionar los dos individuos a ser padres, si no también cuales de la población anterior serán eliminados, para dar espacio a los descendientes.

La diferencia esencial entre el reemplazo generacional y el modelo de estado fijo son las estadísticas de la población que

se recalculan luego de cada cruce y los nuevos descendientes están disponibles inmediatamente para la reproducción. Esto permite al modelo utilizar las características de un individuo prometedor tan pronto como es creado.

Algunos autores dicen que este modelo tiende a evolucionar mucho más rápido que el modelo generacional, sin embargo, en varias investigaciones de Goldberg, se encontraron que las ventajas parecen estar relacionadas con la alta tasa de crecimiento inicial, ellos dicen que los mismos efectos pueden ser obtenidos en rangos de adaptación exponencial o selección por competencia. No encontraron evidencia que este modelo sea mejor que el Generacional.

- **Paralelos.** Parte de la metáfora biológica que motivo a utilizar la búsqueda genética consiste en que es inherentemente paralela, donde al evolucionar se recorren simultáneamente muchas soluciones, cada una representada por un individuo de la población. Sin embargo, es muy común en la naturaleza que no sólo sea una población evolucionando, sino varias poblaciones, normalmente aisladas geográficamente, que originan respuestas diferentes a la presión evolutiva. Esto origina dos modelos que toman en cuenta esta variación, y utilizan no una población como los anteriores sino múltiples poblaciones en forma concurrente.

- **Modelos de Islas.** Si se tiene una población de individuos, ésta se divide en subpoblaciones que evolucionan independientemente como un Algoritmo Genético normal. Ocasionalmente, se producen migraciones entre ellas, permitiéndoles intercambiar material genético.

Con la utilización de la migración, este modelo puede explotar las diferencias en las subpoblaciones; esta variación representa una fuente de diversidad genética. Sin embargo, si un número de individuos emigran en cada generación, ocurre una mezcla global y se eliminan las diferencias locales, y si la migración es infrecuente, es probable que se produzca convergencia prematura en las subpoblaciones.

- **Modelo Celular.** Coloca cada individuo en una matriz, donde cada uno sólo podrá buscar reproducirse con los individuos que tenga a su alrededor (más cerca de casa) escogiendo al azar o al mejor adaptado. El descendiente pasará a ocupar una posición cercana. No hay islas en este modelo, pero hay efectos potenciales similares. Asumiendo que el cruce está restringido a individuos adyacentes, dos individuos separados por 20 espacios están tan aislados como si estuvieran en dos islas, este tipo de separación es conocido como aislamiento por distancia. Luego de la primera evaluación, los individuos están todavía distribuidos al azar sobre la matriz. Posteriormente, empiezan a emerger zonas como cromosomas y adaptaciones semejantes. La reproducción y selección local crea tendencias evolutivas aisladas, luego de varias generaciones, la competencia local resultará en grupos más grandes de individuos semejantes.



## Representación del conocimiento y aprendizaje genético

La representación del conocimiento se hace mediante la codificación de la información e involucra la utilización de árboles de decisión, redes semánticas, reglas, etc. Cada parámetro se representa mediante un conjunto de bits donde cada bit representa un posible estado del parámetro. Por ejemplo, para la regla:

**Si** Cielo = (nublado o lluvioso) y Viento = fuerte **Entonces** Jugar tenis = No

Cielo puede tomar tres valores:

Soleado            Nublado            Lluvioso

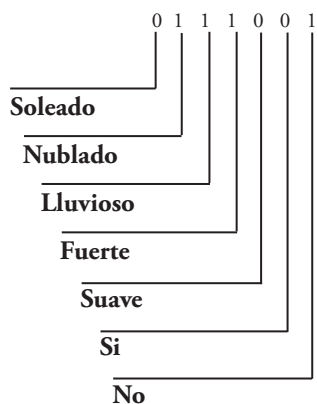
Viento puede tomar 2 valores:

Fuerte            Suave

Y jugar tenis puede tomar 2 valores:

Si            No

Al representar la regla mediante un conjunto de bits, ésta quedaría como lo muestra la figura 4.



**Figura 4.** Cromosoma que representa una regla.

Para representar varios parámetros dentro de las reglas hay que concatenarlos.

Si hay codificaciones válidas pero sintácticamente incorrectas hay que reparar o modificar operadores mediante mutaciones o combinaciones con parámetros correctos.

## Relación pertinente con el entorno

En el entorno productivo de nuestro país es muy difícil encontrar aplicaciones ya realizadas que utilicen algoritmos genéticos como tal, esto debido al desinterés del medio hacia la tecnología. Sin embargo a nivel de investigaciones sí existen varias aplicaciones.

Algunas de las aplicaciones que se pueden realizar son:

\* Programación lineal, programación no lineal y simulación, programación dinámica, etc., en general, problemas de optimización: hallar la ruta más corta, los cuales son muy comunes en la vida real.

\* Bioinformática, para simular procesos biológicos y genéticos que puedan servir para la cura de enfermedades.

\* Aprendizaje computacional (Machine Learning), ya que con ellos se puede adquirir conocimiento que puede ir mejorando el proceso y por ende eliminando el conocimiento irrelevante.

\* Criptografía, aplicación en desarrollo, estos algoritmos permiten construir claves y sistemas de seguridad, que cuando sean vulnerables se mejoren automáticamente.

## CONCLUSIONES

Los algoritmos genéticos se pueden aplicar en diversos campos, debido a que poseen una estructura sencilla y fácil de implementar. Además pueden resolver problemas con un grado de dificultad muy elevado con eficiencia y exactitud y son altamente paralelizables, es decir, que hallan varias soluciones del problema al mismo tiempo.

La aplicación de los algoritmos genéticos al aprendizaje computacional llevan a un buen desarrollo de la Inteligencia Artificial.

Se puede decir que esta área es muy interesante a investigar, ya que es una de las pocas que tiene bastante parentesco con los procesos genéticos y biológicos del ser humano.

---

**BIBLIOGRAFÍA**

BÄCK Thomas, HOFFMEISTER Frank, SCHWEFEL Hans-Paul (1991). A survey of Evolution Strategies. Proceedings of the 4th Int. Conference on Genetic Algorithms, Morgan Kaufman Publishers.

BANZHAF Wolfgang, NORDIN Peter, KELLER Robert E., FRANCONI Frank D. (1998). Genetic Programming. An Introduction. Morgan Kaufmann Publishers, California.

BOOKER L. B., GOLDBERG David E., HOLLAND J. H. (1990). Classifier Systems and Genetic Algorithms. Artificial Intelligence, 40, 235-282p.

BUCKLES B. P., PETRY F. E. (1992). Genetic Algorithms. IEEE Computer Society Press.

DAVIS L. (1991). Handbook of Genetic Algorithms. Van Nostrand Reinhold.

DAWKINS Richard (1994). El gen egoísta. Salvat Ciencia.

FORREST S. (1993). Genetic Algorithms: Principles of Natural Selection Applied to Computation. Science, Vol. 261, No. 5123, 872-878p.

GOLDBERG David E. (1989). Genetic Algorithms in Search, Optimization and Machine Learning. Addison- Wesley Publishing Company, MA.

GRFENTETTE J.J. (1993). Foundations of Genetic Algorithms. Morgan Kaufmann.

HARTIGAN J. A. (1975). Clustering Algorithms. John Wiley, NY.

HOLLAND John (1975). Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, MI.

MICHALEWICZ Zbigniew (1994). Genetic Algorithms + Data Structures = Evolution Programs. Springer-Verlag.

MITCHELL M. (1996). An Introduction to Genetic Algorithms. MIT Press.

KOZA John R. (1992). Genetic Programming. On the Programming of Computers by Means of Natural Selection. The MIT Press.

KOZA John R. (1992b). Genetic Programming II: Automatic Discovery of Reusable Programs. The MIT Press.

SRINIVAS M., PATNAIK L. M. (1994). Genetic Algorithms: A Survey. IEEE Comp