

Algoritmos genéticos. Un ejemplo con hoja de cálculo

Martínez María-Dolores, Soledad María soledad.martinez@upct.es
Bernal García, Juan Jesús juanjesus.bernal@upct.es
Sánchez García, Juan Francisco jf.sanchez@upct.es
Departamento Métodos Cuantitativos e Informáticos
Universidad Politécnica de Cartagena

RESUMEN

Los algoritmos genéticos se utilizan en distintos ámbitos de investigación ya que se basan en la competición entre individuos sobreviviendo aquellos que resultan más fuertes genéticamente. Se pretende trasladar a los alumnos la capacidad que tienen de optimizar problemas reales, realizando un ejemplo sencillo que sirva de base para explicar cual es su funcionamiento inicial, utilizando para ello una herramienta ofimática que pueden utilizar con facilidad tal y como son las hojas de cálculo.

Palabras claves: Algoritmos genéticos; Hoja de Cálculo; Ejemplo Didáctico.

ABSTRACT

The genetic algorithms are used in different research areas as they are based on the competition among individuals to survive those who are genetically strong. It tries to give students the abilities to optimize real problems by a simple example as a basis to explain what their initial operation, using an office automation tool that can be used easily as spreadsheets are.

Keywords: Genetic Algorithms; Spreadsheet; Didactic Example.

Área temática: Metodología y Didáctica

1. INTRODUCCIÓN

Los algoritmos genéticos (AG) han sido y son utilizados para resolver problemas de búsqueda y optimización, asuntos que a su vez pertenecen a áreas de investigación muy diversas, tales como la inteligencia artificial, diseño de materiales o equipamiento industrial y de redes, análisis lingüístico, problemas multicriterio, ingeniería de software, optimización de producción y distribución, descripción de sistemas biológicos, etc.

Se basan en la conocida teoría de la evolución de Darwin, por la que se establece que el conjunto de una población se combina de forma aleatoria tal y como ocurre en la evolución de las especies, para que posteriormente la naturaleza, o en nuestro caso según un criterio establecido, realice una selección de los individuos decidiendo así cuáles sobreviven o se consideran más fuertes respecto al resto y cuáles se quedan por el camino.

El lenguaje utilizado en biología es asumido por la metodología matemática, así que nos encontramos con que el conjunto de soluciones de un problema es lo que se denomina *fenotipo*, mientras que la información que determinan a cada individuo como tal se conoce como *cromosoma*. Si esta información se reescribe en forma binaria y de cadena, entonces nos encontramos con el *genotipo* y a cada uno de los componentes de dicha cadena se le conoce como *gen*. La combinación de los cromosomas iteración tras iteración nos da lo que se estipula como *generaciones*. Una combinación de cromosomas crea una nueva generación, o lo que es lo mismo una *descendencia*.

Fue John Holland¹, un investigador de la Universidad de Michigan a finales de los sesenta, quien se dio cuenta en primer lugar de la potencia de estos cálculos y de las oportunidades que ofrecían a diferentes campos de investigación, por lo que desarrolló una metodología para poder implantarlo en un programa que conseguía que los ordenadores fueran capaces de aprender generación tras generación.

Existen varias definiciones de lo que se considera un algoritmo genético, así que podemos considerar que es un método de búsqueda dirigida basada en probabilidad, o un método de búsqueda de soluciones óptimas mediante la simulación de la evolución

¹ Holland, J.H. "Adaptation in Natural and Artificial Systems"

natural, o un método adaptativo que se usa para resolver problemas de búsqueda y optimización, o bien la que se considera más estricta y que propone John Koza: “*Un algoritmo matemático altamente paralelo que transforma un conjunto de objetos matemáticos 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 de entre las que destaca la recombinación sexual. Cada uno de estos objetos matemáticos suelen 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.*”²

Los AG se encuadran dentro de las técnicas de búsqueda aleatorias y guiadas, dentro de lo que podíamos definir como algoritmos basados en la evolución. Dicha rama se encuentra a la misma altura que la búsqueda tabú, el análisis de simulación o el análisis de redes neuronales. Además muchos los consideran como una parte importante de la Inteligencia Artificial, al mismo nivel que la Programación Genética, la Programación Evolutiva, o las Estrategias Evolutivas, ya que todos ellos se basan en el proceso de evolución de cada individuo y de sus cromosomas, y de cómo sobreviven los que mejor se adaptan y por tanto los que más se reproducen.

2. ¿QUÉ DETERMINA EL USO DE UN ALGORITMO GENÉTICO?

Para saber si es posible usar o no este tipo de técnica, hemos de intentar tener en cuenta que posee una serie de características que lo definen y que a su vez lo limitan:

- Las soluciones están limitadas a un cierto rango porque su espacio de búsqueda es discreto. Se puede usar en espacios de búsqueda continuos cuando el rango de solución sea muy pequeño.
- La función objetivo que marca el problema de optimización a resolver (o *función adaptativa* según su terminología), siempre es maximizada, y tiene que poder ser definida de forma que se nos indique si es buena o no cierta solución, premiando en el primer caso y penalizando en el segundo.

² Koza, J. R. “Genetic Programming. On the Programming of Computers by Means of Natural

- Cada solución va a ser codificada, normalmente en forma binaria.

Los algoritmos genéticos suelen ser eficaces con funciones no derivables, y hay que tener en cuenta que si existen muchos máximos/mínimos locales necesitaremos de más iteraciones para asegurar el local, y que si tiene muchos puntos cercanos al óptimo, no podemos asegurar que encontraremos dicho óptimo, pero sí al menos uno de esos puntos. Esta es una de sus mayores desventajas, aunque empíricamente se ha demostrado que sí que encuentran soluciones muy buenas en un tiempo realmente breve en referencia a otros sistemas de búsqueda de soluciones. Los AG se utilizan por tanto para problemas de muy diversa índole que no tienen una técnica especializada asociada a los mismos, o en su caso en combinación, a otros métodos para mejorar sus resultados iniciales.

3. FUNCIONAMIENTO DE UN ALGORITMO GENÉTICO SIMPLE

Queremos presentar el funcionamiento de un algoritmo genético simple (AGS), también denominado Canónigo³, mediante un ejemplo aplicado a una herramienta de uso común por los estudiantes tal y como es la hoja de cálculo. En general una AGS se compone de las siguientes fases:

1. Codificar la información del problema.
2. Generar aleatoriamente la población inicial. Cada individuo necesita tener codificada su información en forma de cromosoma. Cada uno de estos cromosomas son posibles soluciones del problema a estudiar.
3. Evaluación de la población. A cada cromosoma se le aplica la función de aptitud que ha elegido para su estudio. Mediante esta función asignamos un número real a cada posible solución del problema. Se otorga más probabilidad de emparejamiento a aquellas que sean mejores.
4. Reproducción. Durante el problema se eligen aleatoriamente los individuos que se van a reproducir y el cruce también se hace de forma aleatoria. Dicho cruce se realiza sobre los genotipos de cada pareja elegida como padres. Mediante el operador de

Selection”

cruzamiento sabremos qué parte del padre y de la madre pasa a la descendencia, así como el modo. Holland propone el sistema *crossover* para seleccionar los individuos y decidir cómo cruzarlos, y aunque existen otros sistemas, éste es el más utilizado.

Existen distintos tipos de crossover:

- a) N-puntos: los cromosomas se cortan por n puntos y los genes se intercambian a partir de ahí. El más utilizado es de un punto o de dos puntos.
- b) Uniforme: en un patrón aleatorio de unos y ceros, se intercambian los bits de los dos cromosomas que coincidan donde hay un 1 en el patrón, o se genera un número aleatorio para cada gen, y si supera una probabilidad determinada se intercambia el bit afectado.
- c) Especializados: Se utilizan cuando los mencionados con anterioridad nos dan soluciones inválidas, intentando que la aleatoriedad no influya y se generen así soluciones siempre válidas, asegurando que mantienen la estructura del problema.

5. La Mutación es el siguiente paso ya que va a contribuir a introducir diversidad en el proceso. Tal y como ocurre en la Naturaleza, las mutaciones tendrán una frecuencia baja que habrá que establecer. Consistirá en la alteración de un gen en un individuo determinado. Para generar diversidad también se introducen otras variantes, tales como ampliar la población y la aleatoriedad absoluta en la población inicial.

4. EJEMPLO CON HOJA DE CÁLCULO DE UN AGS

Existen distintos autores que debido a la complejidad de estos métodos se han dedicado a realizar ejemplos sencillos que puedan hacer que el alumno perciba y entienda mejor el alcance y el funcionamiento de la estructura de un algoritmo genético. En esta comunicación nos hacemos eco de uno de esos ejemplos que con alguna variante se repiten en forma de ejemplo en la red, y lo hemos trasladado a un entorno ofimático, para de esta forma enseñarles el uso de la hoja de cálculo desde otra perspectiva. Creemos que es importante que con las herramientas que tienen a su

³ Goldberg lo recogió de lo utilizado por Holland y le incorporó el código binario.

alcance los alumnos se atrevan a modelizar cualquier problema por complicado que éste pueda parecer a priori.

Por este motivo, hemos recogido el ejemplo que plantea Miguel Ángel Muñoz Pérez en su versión de abril de 2005⁴ y lo hemos trasladado tal cual a un fichero de Excel que ahora les presentamos.

4.1. Función adaptativa

Se parte de una función sencilla en este caso de $f(x) = x^2$ y queremos encontrar el valor que la maximiza, pero restringiendo la variable a números enteros entre 0 y 1. La solución es sencilla, ya que el valor máximo de f será 31 al cuadrado y por tanto 961. Como comprobamos, es un ejemplo sencillo que se resuelve sin necesidad de grandes cálculos, pero que sirve para explicar perfectamente cómo funciona un AGS.

	A	B	C	D	E	F	G
1							
2							
3		x=	31				
4		f(x)=	961				
5		Codificación	matriz				
6		0	0	0	0	0	0
7		1	0	0	0	0	1
8		2	0	0	0	1	0
9		3	0	0	0	1	1
10		4	0	0	1	0	0
11		5	0	0	1	0	1
12		6	0	0	1	1	0
13		7	0	0	1	1	1
14		8	0	1	0	0	0
15		9	0	1	0	0	1
16		10	0	1	0	1	0
17		11	0	1	0	1	1
18		12	0	1	1	0	0
19		13	0	1	1	0	1
20		14	0	1	1	1	0
21		15	0	1	1	1	1
22		16	1	0	0	0	0
23		17	1	0	0	0	1
24		18	1	0	0	1	0
25		19	1	0	0	1	1
26		20	1	0	1	0	0
27		21	1	0	1	0	1
28		22	1	0	1	1	0
29		23	1	0	1	1	1
30		24	1	1	0	0	0
31		25	1	1	0	0	1
32		26	1	1	0	1	0
33		27	1	1	0	1	1
34		28	1	1	1	0	0
35		29	1	1	1	0	1
36		30	1	1	1	1	0
37		31	1	1	1	1	1

Figura 1. Codificación de individuos

⁴ <http://taylor.us.es/componentes/miguelangel/algoritmosgeneticos.pdf>

4.2. Codificación y Población de estudio

Vamos a codificar el conjunto de soluciones en forma binaria, para lo cual hacemos combinaciones de 0 y 1 en grupos de 5 genes, y para obtener la x multiplicamos la última componente por 1, la penúltima por 2, la central por 4, la segunda por 8 y la primera por 16, y el resultado de la suma nos dará dicha x (Figura 1). Así obtenemos la población de individuos de nuestro problema a estudiar.

El tamaño de la población a analizar en este ejemplo es de seis individuos que son elegidos de forma aleatoria. En el caso que seguimos nos proponen lanzar una moneda al aire, y si sale cara la primera componente del individuo 1 es un 0 y por el contrario si es cruz, entonces será un 1. Este proceso se repite para los 6 individuos, 5 veces para cada uno de ellos. Nosotros en la hoja de cálculo lo hacemos utilizando la función =Aleatorio.Entre(0;1) tal y como muestran las figuras 2 y 3 respectivamente.

	J	K	L	M	N	O	P
			Tamaño de población:	6			
			Matriz población aleatoria				
		orden	1	2	3	4	5
individuos	1	=ALEATORIO.ENTRE(0;1)	=ALEATORIO.ENTRE(0;1)	=ALEATORIO.ENTRE(0;1)	=ALEATORIO.ENTRE(0;1)	=ALEATORIO.ENTRE(0;1)	=ALEATORIO.ENTRE(0;1)
	2	=ALEATORIO.ENTRE(0;1)	=ALEATORIO.ENTRE(0;1)	=ALEATORIO.ENTRE(0;1)	=ALEATORIO.ENTRE(0;1)	=ALEATORIO.ENTRE(0;1)	=ALEATORIO.ENTRE(0;1)
	3	=ALEATORIO.ENTRE(0;1)	=ALEATORIO.ENTRE(0;1)	=ALEATORIO.ENTRE(0;1)	=ALEATORIO.ENTRE(0;1)	=ALEATORIO.ENTRE(0;1)	=ALEATORIO.ENTRE(0;1)
	4	=ALEATORIO.ENTRE(0;1)	=ALEATORIO.ENTRE(0;1)	=ALEATORIO.ENTRE(0;1)	=ALEATORIO.ENTRE(0;1)	=ALEATORIO.ENTRE(0;1)	=ALEATORIO.ENTRE(0;1)
	5	=ALEATORIO.ENTRE(0;1)	=ALEATORIO.ENTRE(0;1)	=ALEATORIO.ENTRE(0;1)	=ALEATORIO.ENTRE(0;1)	=ALEATORIO.ENTRE(0;1)	=ALEATORIO.ENTRE(0;1)
	6	=ALEATORIO.ENTRE(0;1)	=ALEATORIO.ENTRE(0;1)	=ALEATORIO.ENTRE(0;1)	=ALEATORIO.ENTRE(0;1)	=ALEATORIO.ENTRE(0;1)	=ALEATORIO.ENTRE(0;1)

Figura 2. Población aleatoria

	J	K	L	M	N	O	P	Q
			Tamaño de población:	6				
			Matriz población aleatoria					
		orden	1	2	3	4	5	
individuos	1		1	1	1	0	1	
	2		1	0	1	0	1	
	3		1	0	0	0	1	
	4		0	0	0	0	0	
	5		0	1	0	0	0	
	6		1	0	0	1	0	

Figura 3. Codificación final población elegida

4.3. Selección

Por tanto hemos elegido los 6 siguientes individuos que representan una $f(x)$ determinada: 29, 21, 17, 0, 8 y 18. Ahora deben de competir entre sí, pero antes

calculamos la media (fmed), y observamos cuál es el mejor individuo de los señalados, que en esta ocasión es el número 18 de las x seleccionadas, o el individuo número 1 de nuestra población.

Proceso de selección		
nº individuo	valor de x	valor de f(x)
1	29	841
2	21	441
3	17	289
4	0	0
5	8	64
6	18	324
fmed		326,5
f(x)mejor individuo		841
mejor individuo		1

Figura 4. Mejor de los individuos de la población

Para realizar la selección se hace mediante un enfrentamiento entre dos, estableciendo una pareja para cada individuo de la población, y realizando un torneo entre ellos, de los cuales el mejor genera dos copias de él mismo y el otro queda en el olvido. Esto que a priori parece sencillo, necesita del uso de una tabla intermedia en la hoja de cálculo para asegurarnos de que los enfrentamientos son correctos, es decir que el individuo 1, por ejemplo, desafía a el 2, y a su vez el 2 también se desafía al 1 y no a otro, para lo cual no es suficiente con el uso de números aleatorios, sino que tenemos que introducir la función Jerarquía. En las Figuras 5 y 6 se proponen las fórmulas a utilizar, y en la Figura 7, el resultado de la selección obtenida.

Tabla intermedia			
=AB16	=AB17	=ALEATORIO()	=JERARQUIA(AA16;\$AA\$1
=AB18	=AB19	=ALEATORIO()	=JERARQUIA(AA17;\$AA\$1
=AB20	=AB21	=ALEATORIO()	=JERARQUIA(AA18;\$AA\$1
=Z16	=Y16	=ALEATORIO()	=JERARQUIA(AA19;\$AA\$1
=Z17	=Y17	=ALEATORIO()	=JERARQUIA(AA20;\$AA\$1
=Z18	=Y18	=ALEATORIO()	=JERARQUIA(AA21;\$AA\$1

Figura 5. Tabla intermedia de selección

Podemos comprobar que los individuos finalmente seleccionados son el 29, el 18 y el número 8.

P	Q	R	S	T	U	V	W
5							
=ALEATORI							
=ALEATORI							
=ALEATORI							
=ALEATORI							
=ALEATORI							
=ALEATORI							
selección							
x inicial	jerarquía manual		pareja de torneo		f(x)pareja	Mejor de los dos x del mejor	Elección
=M16	=BUSCARV(O16;Y\$16:\$Z\$21;2,0)	=BUSCAR(Q16;\$O\$16:\$O\$21;\$P\$16:\$P\$21)	=BUSCAR(Q16;\$O\$16:O21;\$P\$16:\$P\$21)	=S16^2	=MAX(T16;N16)	=SI(U16>T16;S16;M16)	=SI(V16=>P16;S16;P16)
=M17	=BUSCARV(O17;Y\$16:\$Z\$21;2,0)	=BUSCAR(Q17;\$O\$16:\$O\$21;\$P\$16:\$P\$21)	=BUSCAR(Q17;\$O\$16:O22;\$P\$16:\$P\$21)	=S17^2	=MAX(T17;N17)	=SI(U17>T17;S17;M17)	=SI(V17=>P17;S17;P17)
=M18	=BUSCARV(O18;Y\$16:\$Z\$21;2,0)	=BUSCAR(Q18;\$O\$16:\$O\$21;\$P\$16:\$P\$21)	=BUSCAR(Q18;\$O\$16:O23;\$P\$16:\$P\$21)	=S18^2	=MAX(T18;N18)	=SI(U18>T18;S18;M18)	=SI(V18=>P18;S18;P18)
=M19	=BUSCARV(O19;Y\$16:\$Z\$21;2,0)	=BUSCAR(Q19;\$O\$16:\$O\$21;\$P\$16:\$P\$21)	=BUSCAR(Q19;\$O\$16:O24;\$P\$16:\$P\$21)	=S19^2	=MAX(T19;N19)	=SI(U19>T19;S19;M19)	=SI(V19=>P19;S19;P19)
=M20	=BUSCARV(O20;Y\$16:\$Z\$21;2,0)	=BUSCAR(Q20;\$O\$16:\$O\$21;\$P\$16:\$P\$21)	=BUSCAR(Q20;\$O\$16:O25;\$P\$16:\$P\$21)	=S20^2	=MAX(T20;N20)	=SI(U20>T20;S20;M20)	=SI(V20=>P20;S20;P20)
=M21	=BUSCARV(O21;Y\$16:\$Z\$21;2,0)	=BUSCAR(Q21;\$O\$16:\$O\$21;\$P\$16:\$P\$21)	=BUSCAR(Q21;\$O\$16:O26;\$P\$16:\$P\$21)	=S21^2	=MAX(T21;N21)	=SI(U21>T21;S21;M21)	=SI(V21=>P21;S21;P21)

Figura 6. Fórmulas de la Selección Final

selección												
x inicial	jerarquía manual		pareja de torneo		f(x)pareja		Mejor de x del mejor		Elección	Tabla intermedia 1		
1	29	2	21	21	441	841	29	29	3	6	0.80494388	3
2	21	1	29	29	841	841	29	29	2	1	0.46405626	6
3	17	6	18	18	324	324	18	18	4	5	0.85590557	2
4	0	5	8	8	64	64	8	8	6	3	0.90153282	1
5	8	4	0	0	0	64	8	8	1	2	0.63399641	4
6	18	3	17	17	289	324	18	18	5	4	0.57810374	5

Figura 7. Selección de individuos

4.4. Cruce y Mutación

El sistema escogido para cruzar a la población seleccionada, es mediante el denominado cruce 1X, es decir formando parejas entre los individuos de una forma aleatoria, al igual que se hace en la selección. Se establece un punto de cruce aleatorio, que consiste en un número aleatorio entre 1, y la longitud del cromosoma del individuo menos uno. En nuestro caso cada individuo tiene 5 componentes binarias, por lo tanto el punto de cruce será un número entre 1 y 4 (Figura 8).

	cruce 1x		longitud individuo - 1
	elección	parejas	cruce
1	29	2	29
2	29	1	29
3	18	6	18
4	8	5	8
5	8	4	8
6	18	3	18

Figura 8. Cruce

Como en el caso anterior, para llegar aquí lo hacemos mediante el uso de una tabla intermedia que nos asegura que las relaciones en el cruce se mantienen en ambas direcciones. Para que el cruce de los individuos sea correcto, realizamos una concatenación de caracteres y escogemos de cada uno de ellos el adecuado, según el punto de corte en función de una búsqueda lógica en la cadena de texto bien por la izquierda, o bien por la derecha de los caracteres asociados a cada individuo (Figuras 9 y 10).

Punto de cruce concatenación (1)	
=P34&Q34&R34&S34&T34	=IZQUIERDA(T26;X34)&DERECHA(T26;5-X34)
=P35&Q35&R35&S35&T35	=IZQUIERDA(T27;X35)&DERECHA(T27;5-X35)
=P36&Q36&R36&S36&T36	=IZQUIERDA(T28;X36)&DERECHA(T28;5-X36)
=P37&Q37&R37&S37&T37	=IZQUIERDA(T29;X37)&DERECHA(T29;5-X37)
=P38&Q38&R38&S38&T38	=IZQUIERDA(T30;X38)&DERECHA(T30;5-X38)
=P39&Q39&R39&S39&T39	=IZQUIERDA(T31;X39)&DERECHA(T31;5-X39)

Figura 9. Concatenación.

Punto de cruce concatenación (1)		Tabla intermedia 2			
10000	10000	4	5	0,67	4
10000	10000	3	6	0,62	5
10000	10000	1	2	0,74	3
00000	00000	5	4	0,33	6
00000	00000	6	3	0,89	1
10000	10000	2	1	0,83	2

Figura 10. Punto de Cruce

Parejas	tabla intermedia punto de cruce
12	1
21	3
36	2
45	4
54	4
63	3

Figura 11. Tabla intermedia 3. Punto de Cruce

En la pareja 1-2 de nuestro ejemplo, el punto de cruce es el 1, según mostramos en la Figura 11, por lo que el padre, es decir el individuo 1 dará el primer gen a su primer hijo de la descendencia y la madre los cuatro restantes, mientras que al segundo hijo entre ambos será al revés.

Esto nos da una nueva generación de individuos que se recogen en la Figura 12, y en la que la *fmed* será mayor, así como la $f(x)$ del mejor de los finalmente seleccionados también será mayor que la primera elegida. Esto implica que los individuos tras el proceso de selección y cruce son mejores. Podemos iterar el proceso, volviendo a realizar la selección y el cruce, tomando éstos últimos individuos como población inicial. Las iteraciones nos aproximarán cada vez más al óptimo, aunque como ya hemos dicho no lo garantizan, sí que nos indican una solución muy buena para nuestro problema inicial.

Tal y como indica Muñoz Pérez junto con otros autores, podemos incurrir en un problema de homogeneización es decir que los individuos al final sean prácticamente idénticos, y más en un problema con una población inicial tan pequeña, por lo que podemos quedar estancados y no conseguir optimizar nuestra función. Para solucionar esto, ya hemos indicado que podemos incluir cierta diversidad mediante la mutación, escogiendo al azar un número determinado de genes de la población que nos queda tras un cruce, y alterar su codificación de manera aleatoria.

5. CONCLUSIONES

Como ya hemos mencionado, pretendemos con este trabajo poner al alcance del alumno un ejemplo de los que circulan por la red sobre algoritmos genéticos, planteándolo desde una sencilla explicación de qué son y en qué se utilizan, y trasladándolo al uso de una hoja de cálculo. Además, creemos que este sistema permite que los alumnos puedan visualizar mediante una función sencilla cómo se compone y en qué consiste una modelización de un AGS cualquiera, ya que dada su complejidad pueden resultar más complicados de comprender por el alumno sin el uso de dicha herramienta. Por otro lado, este ejemplo nos permite también ahondar en el uso de la simulación así como de funciones lógicas con hoja de cálculo, por esta razón se pretende su incorporación al grupo de ejercicios prácticos de la asignatura de

Informática de Gestión correspondiente al Grado en Ciencias de la Empresa que se imparte en nuestra Universidad.

6. REFERENCIAS BIBLIOGRÁFICAS

- GOLDBERG, D.E. (1989), “Genetic Algorithms in Search, Optimization and Machine Learning”. Addison-Wesley Publishing Company, 412 p.
- HOLLAND, J.H. (1975) “Adaptation in Natural and Artificial Systems”. University of Michigan Press, 211 p.
- KOZA, J.R. (1992), “Genetic Programming. On the Programming of Computers by Means of Natural Selection”. The MIT Press, 819 p.
- KURI, A. (2000) “Algoritmos Genéticos”. Centro de Investigación en Computación de México. <http://redalyc.uaemex.mx/redalyc/pdf/615/61530308.pdf>
<http://cursos.itam.mx/akuri/PUBLICA.CNS/2000/AGS.PDF>
- MARCZYK, A. (2004) “algoritmos genéticos y computación evolutiva” <http://the-geek.org/docs/algen/>
- MERELO GUERVÓS, J.J (consultado en mayo de 2010)“Informática evolutiva: Algoritmos genéticos” <http://geneura.ugr.es/~jmerelo/ie/ags.htm>
- MOSCATO, P. y Cota, C. (consultado en mayo de 2010) “A Gentle Introduction to Memetic Algorithms” <http://citeersx.ist.psu.edu>
- MUÑOZ PÉREZ, M.A. (2005). “Algoritmos genéticos”. <http://taylor.us.es/componentes/miguelangel/algoritmosgeneticos.pdf>
- VAREDas, F.J y VICO, F.J (Computación Evolutiva Basada en un modelo de codificación implícita”. Inteligencia Artificial, 5, pág. 20-25