

<h1>GIAR</h1>	Universidad Tecnológica Nacional Facultad Regional Buenos Aires Introducción a la Inteligencia Artificial	Fecha de creación 04/06/2009 10:06:00 a.m.
	<h2>Algoritmos Geneticos</h2> <p>C. Verrastro, J.C. Gómez, R. Alcoberro</p>	Fecha Última Rev. 04/06/2009 04:50:00 p.m.
		Revisión 2 Página 1 de 9

Índice

1	Introducción.....	1
1.1	Estructura básica de un Programa Evolutivo.....	3
2	Algoritmos Genéticos. ¿Qué son?.....	3
2.1	Optimización de una función simple.....	4
2.2	Población Inicial.....	4
2.3	Función de evaluación.....	4
2.4	Operadores genéticos.....	5
3	Algoritmos Genéticos ¿Cómo trabajan?.....	5
4	Algoritmos Genéticos ¿Por qué funcionan?.....	7
4.1	Orden del esquema.....	7
4.2	Longitud del esquema.....	8
5	Teorema del esquema.....	9
6	Hipótesis de los bloques constructivos.....	9
7	Bibliografía:.....	9

1 *Introducción*

Durante los últimos treinta años ha crecido el interés en sistemas de resolución de problemas basados en los principios de evolución y herencia. Estos sistemas mantienen una población de soluciones potenciales transformando algunos de estos individuos por medio de operadores "genéticos" y usando criterios de evaluación que determinan la "adaptación" de los individuos y algoritmos de "selección" de los más aptos.

Esta técnica se basa en los mecanismos de selección que utiliza la naturaleza, de acuerdo a los cuales los individuos más aptos de una población son los que sobreviven, al adaptarse más fácilmente a los cambios que se producen en su entorno. Hoy en día se sabe 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 (por ejemplo los que le permiten a un individuo adaptarse mejor a su entorno) del mismo se transmiten a sus descendientes, cuando éste se reproduce.

Un investigador de la Universidad de Michigan llamado John Holland era consciente de la importancia de la selección natural, y a fines de los 60s desarrolló una técnica que permitió incorporarla en un programa de computadora. Su objetivo era lograr que las computadoras aprendieran por sí mismas. La técnica que inventó Holland se llamó originalmente "planes reproductivos", pero se hizo popular recién en 1975.

Una definición bastante completa de un algoritmo genético es la siguiente:

Es un algoritmo matemático altamente paralelizable que maneja un conjunto finito de soluciones individuales transformándolas con operaciones modeladas de acuerdo al principio Darwiniano de reproducción y supervivencia del más apto. Utiliza mecanismos inspirados en los sistemas biológicos, es decir una serie de operaciones genéticas de entre las que destaca la recombinación sexual, la mutación y la selección.

Cada una de las soluciones pueden ser vistas como objetos matemáticos y suelen ser una cadena de caracteres (letras o números) de longitud fija que se ajusta al modelo

<h1>GIAR</h1>	Universidad Tecnológica Nacional Facultad Regional Buenos Aires Introducción a la Inteligencia Artificial	Fecha de creación 04/06/2009 10:06:00 a.m.
	<h2>Algoritmos Geneticos</h2> C. Verrastro, J.C. Gómez, R. Alcoberro	Fecha Última Rev. 04/06/2009 04:50:00 p.m.
		Revisión 2 Página 2 de 9

de las cadenas de cromosomas, y se les asocia con una cierta función matemática que refleja su aptitud.

La aplicación más común de los algoritmos genéticos ha sido la solución de problemas de optimización y búsqueda, en donde han mostrado ser muy eficientes y confiables. Sin embargo, no todos los problemas pueden ser resueltos por esta técnica. Se recomienda en general tomar en cuenta algunas de las características del problema a resolver antes de intentar usar algoritmos genéticos para resolverlo:

El espacio de búsqueda, es decir, las posibles soluciones, debe estar delimitado dentro de un cierto rango. Debe ser un espacio finito.

Se debe poder definir una función de aptitud que nos indique qué tan buena o mala es una cierta solución. Se debe poder medir la aptitud del individuo (solución).

Las soluciones deben codificarse de una forma que resulte relativamente fácil de implementar en una computadora.

El primer punto es muy importante, se recomienda resolver problemas que tengan espacios de búsqueda discretos, aunque éstos sean muy grandes. Sin embargo, también puede intentarse usar la técnica con espacios de búsqueda continuos, pero preferentemente cuando exista un rango de soluciones relativamente pequeño.

La función de aptitud es la función objetivo de nuestro problema de optimización. El algoritmo genético intenta maximizarla. También se puede utilizar una función que se deba minimizar (como una función de error) debe cuidarse, por supuesto, que el recíproco de la función no genere una división por cero. Una característica que debe tener esta función es que debe ser capaz de "castigar" a las malas soluciones, y de "premiar" a las buenas, de forma que sean estas últimas las que se propaguen con mayor rapidez.

La codificación más común de las respuestas es a través de cadenas binarias, aunque se han utilizado también números reales y letras. El primero de estos esquemas ha gozado de mucha popularidad debido a que es el que propuso originalmente Holland, y además porque resulta muy sencillo de implementar.

Entre las ventajas de utilizar algoritmos genéticos se pueden nombrar:

No necesitan conocimientos específicos sobre el problema que intentan resolver.

Operan de forma simultánea con varias soluciones, en vez de trabajar de forma secuencial como las técnicas tradicionales.

Cuando se usan para problemas de optimización (maximizar una función objetivo) resultan menos afectados por los máximos locales (falsas soluciones) que las técnicas tradicionales.

Resulta sumamente fácil ejecutarlos en arquitecturas multiprocesador, en forma paralela.

Usan operadores probabilísticos, en vez de los típicos operadores determinísticos de las otras técnicas.

Algunos de los posibles problemas son:

Pueden tardar mucho en converger, o no converger en absoluto, dependiendo en cierta medida de los parámetros que se utilicen (tamaño de la población, número de generaciones, características de la función objetivo, etc.).

Pueden converger prematuramente debido a una serie de problemas de diversa índole.

<h1>GIAR</h1>	Universidad Tecnológica Nacional Facultad Regional Buenos Aires Introducción a la Inteligencia Artificial	Fecha de creación 04/06/2009 10:06:00 a.m.
	<h2>Algoritmos Geneticos</h2> C. Verrastro, J.C. Gómez, R. Alcoberro	Fecha Última Rev. 04/06/2009 04:50:00 p.m.
		Revisión 2 Página 3 de 9

1.1 Estructura básica de un Programa Evolutivo.

```

begin
t = 0
inicializar P(t)
evaluar P(t)
  While (not condición de terminación) do
  begin
    t = t + 1
    seleccionar P(t) de P(t-1)
    alterar P(t)
    evaluar P(t)
  end
end
end

```

Un programa evolutivo es un algoritmo probabilístico que mantiene una población $p(t) = \{x_1, \dots, x_n\}$ para la iteración t .

Cada individuo representa una solución potencial del problema considerado. Cada solución es evaluada para obtener una medida de su aptitud. Luego se forma una nueva población (iteración $t+1$) seleccionando los individuos más aptos. Algunos de los nuevos individuos se someten a transformaciones aplicándoles operadores genéticos para generar nuevas soluciones (individuos).

Existen transformaciones unarias (mutación) que crean un nuevo individuo a partir de producirle pequeños cambios. Existen otras transformaciones, como la operación de cruce, que crean un nuevo individuo combinando partes de dos o más individuos. Después de algún número de generaciones se espera que el algoritmo converja a un estado donde el individuo más adaptado representa una solución sub-óptima del problema considerado.

El algoritmo genético clásico utiliza un número binario de longitud fija para sus individuos (como estructura de sus cromosomas) y dos operadores: mutación binaria y cruce binaria.

Los métodos de algoritmo genético son clasificados como métodos débiles pero en presencia de restricciones no triviales pueden migrar rápidamente a métodos fuertes.

2 Algoritmos Genéticos. ¿Qué son?

La metáfora subyacente en los algoritmos genéticos es la evolución natural. En la evolución cada especie se modifica en la búsqueda de adaptaciones benéficas a un ambiente complicado y cambiante. El conocimiento de cada especie está incorporado a la estructura cromosómica de sus miembros.

Por esto los algoritmos genéticos incorporan la terminología de la genética natural y se usan principalmente en problemas de optimización y búsqueda de máximos (o mínimos) pero no están limitados exclusivamente a esta clase de problemas.

En forma más general se los puede definir como algoritmos de búsqueda estocástica multidireccional.

<h1>GIAR</h1>	Universidad Tecnológica Nacional Facultad Regional Buenos Aires Introducción a la Inteligencia Artificial	Fecha de creación 04/06/2009 10:06:00 a.m.
	<h2>Algoritmos Geneticos</h2> C. Verrastro, J.C. Gómez, R. Alcoberro	Fecha Última Rev. 04/06/2009 04:50:00 p.m.
		Revisión 2 Página 4 de 9

Para un problema particular deben tener los siguientes 5 componentes:

- Una representación genética de las soluciones potenciales.
- Una forma de crear la población inicial.
- Una función de evaluación que juegue el rol del ambiente clasificando los individuos en función de su adaptación.
- Operadores genéticos que alteren la composición de los hijos.
- Valores de los parámetros para las distintas variables que usa el algoritmo (tamaño de la población, probabilidad de aplicación de los operadores genéticos etc.)

2.1 Optimización de una función simple

Consideremos la siguiente función

$$f(x) = x * \text{sen}(10 * \pi * x) + 1$$

El problema es encontrar x en el rango de [-1 a 2] que maximize el valor de f(x).

Representación

Se usará un vector binario que represente el valor de la variable x dentro del rango deseado. Su longitud dependerá de la precisión requerida, por ejemplo 6 lugares decimales.

Para calcular el tamaño del vector binario de cada cromosoma se divide el espacio de búsqueda por la precisión que se desea alcanzar y se toma el valor entero superior.

Entonces tenemos:

Precisión: 0.000001

Espacio de búsqueda: $2 - (-1) = 3$

Esto significa que 22 bits serán suficientes para representar un número comprendido entre los valores -1,000000 y 2,000000.

Para saber a que valor de X corresponde un determinado vector binario hacemos:

$$x = -1 + x_d \cdot 3 / (2^{22} - 1)$$

Donde x_d es el valor decimal del vector binario.

2.2 Población Inicial

En este caso se parte de una población inicial donde cada uno de los 22 bits de cada individuo es inicializado al azar.

2.3 Función de evaluación

Para este caso simple la función de evaluación es equivalente a la función.

$$\text{eval}(V) = f(x) = x * \text{sen}(10 * \pi * x) + 1$$

<h1>GIAR</h1>	Universidad Tecnológica Nacional Facultad Regional Buenos Aires Introducción a la Inteligencia Artificial	Fecha de creación 04/06/2009 10:06:00 a.m.
	<h2>Algoritmos Genéticos</h2> C. Verrastro, J.C. Gómez, R. Alcoberro	Fecha Última Rev. 04/06/2009 04:50:00 p.m.
		Revisión 2 Página 5 de 9

2.4 Operadores genéticos

- Selección
- Cruza
- Mutación

Selección

Mediante la Selección se eligen los individuos de la generación anterior que pasarán a la siguiente. La cantidad de individuos está predeterminada por el tamaño de la población. Para ello, en la forma mas simple de selección se usa una ruleta ponderada.

Mutación

La mutación cambia uno o más genes del cromosoma (bits del vector binario) con probabilidad igual a la tasa de mutación.

Cruza

Para la cruza, se seleccionan al azar dos cromosomas padres y un punto de fraccionamiento también al azar y se obtienen dos hijos permutando los fragmentos de ambos padres.

$$P_1 = (a,b) \quad P_2 = (c,d) \quad \rightarrow \quad H_1 = (a,d) \quad H_2 = (c,b)$$

Parámetros

Se deben definir los siguientes parámetros

Tamaño de población = 50

Probabilidad de cruza = 0,25

Probabilidad de mutación = 0,01

3 Algoritmos Genéticos ¿Cómo trabajan?

Sin pérdida de generalidad podemos suponer que el problema de optimización es un problema de maximización ya que minimizar $f(x)$ es equivalente a maximizar $g(x)$ donde $f(x) = -g(x)$

Además suponemos que la función adopta sólo valores positivos, de no ser así adicionamos una constante.

Si se quiere maximizar una función de k variables $f(x_1, \dots, x_k)$ donde cada x_i toma valores del un dominio $D_i = [a_i, b_i]$ entonces para una dada precisión cada variable. Será representada por un vector binario de longitud fijada por el número de cifras significativas requerido, de forma análoga al ejemplo anterior.

En este caso, el cromosoma (individuo) será representado por un número binario de longitud $m = m_1 + \dots + m_k$ donde m_i es la longitud del vector binario que representa a x_i

<h1>GIAR</h1>	Universidad Tecnológica Nacional Facultad Regional Buenos Aires Introducción a la Inteligencia Artificial	Fecha de creación 04/06/2009 10:06:00 a.m. Fecha Última Rev. 04/06/2009 04:50:00 p.m.
	<h2>Algoritmos Geneticos</h2> C. Verrastro, J.C. Gómez, R. Alcoberro	Revisión 2 Página 6 de 9

La población inicial se puede crear al azar o bien usar la información disponible acerca del problema.

Para la evaluación se usa la misma función y para el proceso de selección se crea una ruleta con ranuras normalizadas con respecto al valor de la función de evaluación, en otras palabras

- Calcule la aptitud, con $eval(v_i)$, de cada cromosoma v_i
- Encuentre la adaptación total F de la población como la suma de las adaptaciones individuales
- Calcule la probabilidad de selección p_i de cada cromosoma como:

$$p_i = eval(v_i)/F$$
- Calcule la probabilidad acumulada q_i para cada cromosoma como:

$$q_i = \sum_{j=1, i} p_j$$

Ejemplo:

Cromosoma N°	Cadena	Aptitud	% del Total	% Acumulativo
1	11010110	254	24.5	24.5
2	10100111	47	4.5	29
3	00110110	457	44.1	73.1
4	01110010	194	18.7	91.8
5	11110010	85	8.2	100
Total		1037	100.0	

El proceso de selección se basa en hacer girar la ruleta tantas veces como número de individuos T tenga la población, por cada giro se selecciona un cromosoma para la nueva población de acuerdo al siguiente criterio:

- Genere un número al azar r en el rango de $[0, \dots, 1]$
- Si $r < q_i$ entonces seleccione el primer cromosoma (v_1). De otra manera seleccione el cromosoma v_i que cumpla con la condición $q_{i-1} < r < q_i$

Obviamente un mismo cromosoma puede ser seleccionado mas de una vez. Los mejores obtienen mas copias, el promedio permanece igual, y los peores mueren.

Sobre esta nueva población se aplica el operador de cruce del siguiente modo, para cada cromosoma

- Genere un numero al azar r en el rango de $[0, \dots, 1]$
- Si $r < p_c$ (prob. de cruce) entonces seleccione el cromosoma en cuestión
- Repetir el paso anterior hasta encontrar otro cromosoma para realizar la combinación
- Para cada pareja determine al azar un punto de fraccionamiento
- Genere dos hijos de acuerdo a la ley de formación

<h1>GIAR</h1>	Universidad Tecnológica Nacional Facultad Regional Buenos Aires Introducción a la Inteligencia Artificial	Fecha de creación 04/06/2009 10:06:00 a.m. Fecha Última Rev. 04/06/2009 04:50:00 p.m.
	<h2>Algoritmos Genéticos</h2> <p>C. Verrastro, J.C. Gómez, R. Alcoberro</p>	Revisión 2 Página 7 de 9

$P_1 = (a,b)$ $P_2 = (c,d)$ -----> $H_1 = (a,d)$ $H_2 = (c,b)$

Ejemplo:

$P_1 = 01111011$ $P_2 = 10010100$
 $H_1 = 01111100$ $H_2 = 10010011$

- Reemplace los padres por los hijos en la nueva población

A la población así conformada se le aplica el operador unario de mutación sobre a cada cromosoma seleccionado y bit a bit.

El criterio de selección para la mutación es el siguiente:

Para cada cromosoma de la generación actual y para cada bit del cromosoma:

- Genere un número al azar r en el rango de $[0, \dots, 1]$
- Si $r < p_m$ (prob. de mutación) complemente el bit.

Finalizado este proceso la población esté en otro ciclo evolutivo (generación) que comienza con la evaluación de la aptitud de cada individuo al ambiente

4 Algoritmos Genéticos ¿Por qué funcionan?

Los fundamentos teóricos de los algoritmos genéticos descansan sobre la representación binaria de las soluciones y en la noción de esquema.

Un esquema se construye introduciendo el símbolo * (Indiferente) en el alfabeto de los genes, por ej., si consideramos el esquema de longitud 10 $\{(*111100100)\}$ coincide con dos vectores binarios $\{(1111100100), (0111100100)\}$

Esta claro que cada esquema coincidirá con 2^r vectores, donde r es el número de * del esquema. Por otro lado cada vector de longitud m coincidirá con 2^m esquemas distintos, de los 3^m esquemas de longitud m posibles (Es un sistema ternario: 0, 1, *). En una población de n individuos pueden representarse entre 2^m y $n \cdot 2^m$ esquemas, siendo 2^m cuando todos los individuos son iguales y $n \cdot 2^m$ cuando son todos distintos.

Los esquemas tienen dos propiedades, orden y longitud definida, y sobre ellas se formula el teorema del esquema.

4.1 Orden del esquema

$O(S)$ es el número de 0 y de 1 del esquema sin contar los * (el número de posiciones fijas) y representa el grado de especialización del esquema

$$S1 = (**001*110) \quad O(S1)=6$$

El orden del esquema es útil para calcular la probabilidad de supervivencia del esquema frente a las mutaciones.

<h1>GIAR</h1>	Universidad Tecnológica Nacional Facultad Regional Buenos Aires Introducción a la Inteligencia Artificial	Fecha de creación 04/06/2009 10:06:00 a.m.
	<h2>Algoritmos Geneticos</h2> C. Verrastro, J.C. Gómez, R. Alcoberro	Fecha Última Rev. 04/06/2009 04:50:00 p.m.
		Revisión 2 Página 8 de 9

4.2 Longitud del esquema

La longitud definida ($d(s)$) de un esquema es la distancia entre la primera y la última posición fija del esquema.

$$d(S1) = 10 - 4 = 6$$

Este parámetro está relacionado con la noción de compactación de la información contenida en el esquema y es útil para el cálculo de supervivencia del esquema frente al operador de cruce.

Otra propiedad de los esquemas es su valor de adaptación $eval(S,t)$ y está definida como la adaptación promedio de todos los cromosomas que coinciden con el esquema S .

Un cromosoma V_i tiene la probabilidad $eval(V_i,t)/F(t)$ de ser seleccionado en una única selección.

Si en una población de T individuos, el número de cromosomas que coincide con el esquema S es $e(S,t)$ entonces en la generación siguiente el número de cromosomas que coinciden con S será.

$$e(S,t+1) = e(S,t) \cdot T \cdot eval(S,t) / F(t)$$

Si se introduce el concepto de adaptación promedio de la población como

$$FM(t) = F(t)/T$$

entonces

$$e(S,t+1) = e(S,t) \cdot eval(S,t)/FM(t)$$

Esta es la llamada ecuación de crecimiento reproductivo.

Esto significa que el número de individuos que coinciden con esquemas con aptitud superior al promedio crece exponencialmente con t y el número de individuos que coinciden con esquemas con aptitud inferior a la media decrecen exponencialmente. Pero la selección por sí sola no introduce nuevos individuos, solo copia los existentes. Así la recombinación tiene la responsabilidad de introducir nuevos individuos a la población.

La probabilidad de supervivencia de un esquema frente a la cruce es

$$p_s(S) = 1 - p_c \cdot d(s)/(m - 1)$$

La probabilidad de supervivencia de un esquema frente a la mutación es

$$p_s(S) = 1 - O(S) \cdot p_m$$

Combinando los efectos de la cruce y la mutación en la ecuación de crecimiento y considerando: $p_c \ll 1$ y $p_m \ll 1$

$$e(S,t+1) = e(S,t) \cdot eval(S,t)/FM(t) \cdot [1 - p_c \cdot d(s)/(m - 1) - O(s) \cdot p_m]$$

<h1>GIAR</h1>	Universidad Tecnológica Nacional Facultad Regional Buenos Aires Introducción a la Inteligencia Artificial	Fecha de creación 04/06/2009 10:06:00 a.m.
	<h2>Algoritmos Geneticos</h2> C. Verrastro, J.C. Gómez, R. Alcoberro	Fecha Última Rev. 04/06/2009 04:50:00 p.m.
		Revisión 2 Página 9 de 9

5 *Teorema del esquema*

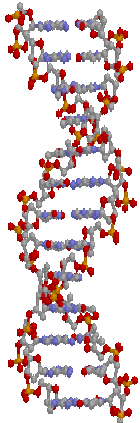
Esquemas cortos, de pequeño orden, con aptitud superior al promedio reciben intentos exponencialmente crecientes en generaciones subsecuentes de un algoritmo genético

6 *Hipótesis de los bloques constructivos*

Un algoritmo genético muestra un funcionamiento cuasi óptimo por medio de la superposición de esquemas cortos, de pequeño orden y significativos, llamados bloques constructivos.


7 *Bibliografía:*

- <http://www.econ.iastate.edu/tesfatsi/holland.GAIntro.htm>
- Genetic Algorithms + Data Structures = Evolution Programs. Zbigniew Michalewicz
- Fuzzy and Neural Approaches in Engineering. Tsoukalas, Lefteri H. Wiley Interscience.
- Inteligencia Artificial, un enfoque moderno. Russell y Norvig. Editorial Pearson.
- <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.29.9772&rep=rep1&type=pdf>



Introducción a los algoritmos genéticos y sus aplicaciones


Juan Carlos Gómez
 Claudio Verrastro
 Rodrigo Alcoberro
 Grupo de Inteligencia Artificial y Robótica.
 UTN-FRBA
 Versión 4



Temario


- Evolución.
- Sistemas evolutivos, paralelos con la naturaleza
- Algoritmos genéticos y Programación Evolutiva.
- Los Algoritmos Genéticos como método de optimización y búsqueda de soluciones
- Teorema del esquema

Algoritmos Genéticos 2



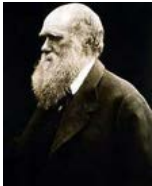
Evolución

Jean Baptiste Lamarck (1744-1829)
 Charles Darwin (1809-1882)
 Alfred Russel Wallace (1823-1913)




“On the Origin of Species” 24 de nov. de 1859)

- La gran variedad de organismos existentes evolucionaron de formas más simples.
- Adaptación al medio ambiente cambiante.
- Herencia de características.
- Selección natural.



Charles Darwin

Algoritmos Genéticos 3




Qué dijo Darwin...

El Origen de las especies (1859)

“Existen organismos que se reproducen y la progenie hereda características de sus progenitores, existen variaciones de características. Si el medio ambiente no admite a todos los miembros de una población en crecimiento, entonces aquellos miembros de la población con características menos adaptadas (según lo determine su medio ambiente) morirán con mayor probabilidad. Entonces aquellos miembros con características mejor adaptadas sobrevivirán más probablemente”.


Algoritmos Genéticos 4



Evolución (premisas)

- Estas características se transmiten con variaciones a la progenie.
- Esa variabilidad otorga ventajas/desventajas para la supervivencia y la reproducción.
- Estas características se pueden extender entre la población con el paso de las generaciones.

Algoritmos Genéticos 5




Genotipo, fenotipo y ambiente

Las características de un ser vivo se almacenan codificadas en los genes de su ADN. Genotipo.

Fenotipo es la expresión de los genes modificado por el ambiente.


Fenotipo = genotipo + ambiente



Proceso evolutivo está controlado por el ambiente y los genes que posee el individuo.

Algoritmos Genéticos 6


Individuo, población, generación




- El individuo debe ser capaz de reproducirse.
- Debe existir una población de estos individuos.
- La variedad de los individuos potencia el proceso evolutivo (importancia de la biodiversidad)
- El proceso lleva muchas generaciones

Algoritmos Genéticos 7

Proceso evolutivo




Está controlado por el ambiente y los genes que posee el individuo



Algoritmos Genéticos 8

Algoritmos genéticos




- Un AG es una **representación computacional** utilizada en la **búsqueda de soluciones** en la que se aplican mecanismos similares a los de la evolución y donde sobreviven los **individuos más aptos**


Nota: Ver John Holland (70)

Algoritmos Genéticos 9


Mecanismos de los AG




Selección



Cruza




Mutación



Algoritmos Genéticos 10


Algoritmos genéticos



- Se utiliza principalmente en problemas de optimización y búsqueda
- El espacio de búsqueda puede ser grande pero debe ser finito
- Las soluciones deben poder codificarse


Algoritmos Genéticos 11

Algoritmos genéticos



- Utiliza una función objetivo que mide la "aptitud" de los individuos
- Se realiza la búsqueda en una población de individuos y no en uno a la vez
- Es un método sub-óptimo y probabilístico


Algoritmos Genéticos 12



Algoritmos genéticos

- Aplica la evolución a los cromosomas
- Los cromosomas se codifican en **cadenas de bits**
- Algunos problemas no se pueden mapear en forma sencilla a los cromosomas


Algoritmos Genéticos 13



Algoritmos genéticos

```
graph TD; Problema --> ProblemaModificado[Problema modificado  
Mapping  
Bit string]; AlgoritmoGenetico[Algoritmo genético] --> ProblemaModificado;
```


Algoritmos Genéticos 14



Programación evolutiva

```
graph TD; Problema --> Programa[Programa  
Building Blocks]; AlgoritmoGenetico[Algoritmo genético] --> Programa;
```

Algoritmos Genéticos 15




Programación evolutiva

- Las posibles soluciones no se codifican en cadenas de bits
- Se transforma el Algoritmo genético para resolver el problema
- Se trabaja con “Building Blocks” (módulos de programa) que se agregan y combinan para resolver la tarea

Nota: Ver trabajos de John R. Koza y Peter Nordin


Algoritmos Genéticos 16



Algoritmos Genéticos. Descripción del funcionamiento

- Cromosoma
 - Definición
 - Representación numérica
- Operadores genéticos
 - Selección
 - Crossover (cruza)
 - Mutación

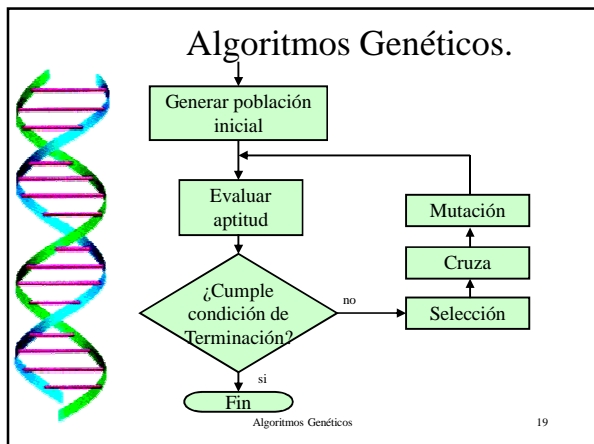
Algoritmos Genéticos 17




Algoritmos Genéticos. Descripción del funcionamiento

- Mapeo del genotipo al fenotipo
- Ejemplos prácticos de uso

Algoritmos Genéticos 18






Cromosoma

- Llamado también individuo o genotipo
- Contiene la información de la solución del problema en forma codificada
- La codificación mas usual es la cadena de bits

1110001010010101


Algoritmos Genéticos 20



Operadores genéticos.


Selección

- Obtiene los mejores cromosomas de la población inicial
- Se utiliza una “ruleta” ponderada en donde cada porción representa un individuo
- En general se mantiene el tamaño de la población constante



Algoritmos Genéticos 21

Operadores genéticos. Selección




En forma de algoritmo:
 $r = \text{rand}()$
 r: número aleatorio entre 0 y 1
 q(i) es la probabilidad acumulada del cromosoma iésimo

- Si $r < q(1)$ entonces se selecciona el primer cromosoma $v(1)$
- Si $q(i-1) < r < q(i)$ se selecciona el cromosoma $v(i)$

Algoritmos Genéticos 22

Operadores genéticos. Selección




- Ejemplo:

Cromosoma No	Cadena	Aptitud	% del Total	% Acumulativo
1	11010110	254	24.5	24.5
2	10100111	47	4.5	29
3	00110110	457	44.1	73.1
4	01110010	194	18.7	91.8
5	11110010	85	8.2	100
Total		1037	100.0	

- Si $r = 0,35$ se selecciona el individuo 3

Algoritmos Genéticos 23

Operadores genéticos. Crossover (Cruza)



- Se decide en forma aleatoria si un determinado individuo se debe cruzar (probabilidad de crossover)
- Se combinan los bits de los progenitores para conformar el individuo descendiente
- Los modos mas usuales son de recombinación a partir de un punto o entre dos puntos

Algoritmos Genéticos 24

Operadores genéticos. Crossover (Cruza)

- El punto de cruce se obtiene de forma aleatoria

$$n = \text{ceil}((nbits - 1) \cdot \text{rand}())$$

Padres: 11001010 01011110

Hijos: 11001110 01011010

Algoritmos Genéticos 25

Operadores genéticos. Crossover (Cruza)

- Cuando se trata de dos puntos tenemos:

Padres: 11001010 01011110

Hijos: 01001110 11011010

Algoritmos Genéticos 26


Operadores genéticos. Mutación

- Se tratan todos los individuos como un único string
- Se aplica la probabilidad de mutación a cada bit y si $\text{rand}() < \text{pm}(i)$ se muta el bit i

...01001110... → ...01000110...

Algoritmos Genéticos 27

Mapeo del cromosoma




- Define la relación entre los bits del cromosoma y el problema de aplicación
- Si se trata de números el mapeo es trivial
- En algunos casos puede complicarse

8 → 00001000

Algoritmos Genéticos 28


Forma de implementación



- Se calcula la aptitud (fitness) de cada cromosoma:
 $f(i) = \text{eval}(v(i)) ; i=1..pop_size$
- Se obtiene el valor de la aptitud de toda la población:
 $F = \text{Sum}(\text{eval}(v(i))); i=1.. pop_size$
- Se calcula la probabilidad de selección de cada cromosoma:
 $p(i) = \text{eval}(v(i)) / F$


Algoritmos Genéticos 29

Forma de implementación



- Se calcula la probabilidad acumulada de cada cromosoma:
 $q(i) = \text{Sum}(p(j)); j=1.. i$
- Se seleccionan aleatoriamente los individuos de acuerdo a la probabilidad acumulada
- Se obtiene la nueva población


Algoritmos Genéticos 30



Forma de implementación

- Luego, por cada cromosoma, se genera nuevamente un número r aleatorio entre 0 y 1
 si $r < \text{Probabilidad de Crossover}$
 se selecciona al cromosoma para cruzar
- Se genera r nuevamente para definir el punto de cruce y se efectúa la combinación.


Algoritmos Genéticos 31



Forma de implementación

- Se genera r nuevamente para aplicar la mutación
 si $r < \text{Prob de mutación}$
 se selecciona al bit para mutar
- Se realiza esto bit a bit a todos los individuos

Algoritmos Genéticos 32



Ejemplo: Optimización de una función

- Ejemplo de uso de los AG para encontrar el máximo de la función


$$f(x) = x \cdot \sin(10 \cdot \text{PI} \cdot x) + 1$$

$$-1 < x < 2$$
- Se desea tener una precisión de 6 decimales (10^{-6}).
- Debemos dividir el espacio de búsqueda tal que respete la precisión.

$$(2 - (-1)) \cdot 10^6 = 3000000$$

Algoritmos Genéticos 33


Ej. Optimización de una función



- Usamos una codificación binaria
 $2^N > 3\,000\,000 \Rightarrow N = 22$
- Se realizan los pasos indicados anteriormente

Algoritmos Genéticos 34

Ej. Algoritmo




```

begin
t = 0
inicializar P(t)
evaluar P(t)
While (not condición de terminación) do
begin
t = t + 1
seleccionar P(t) de P(t-1)
cruzar P(t)
mutar P(t)
evaluar P(t)
end
end
    
```

Algoritmos Genéticos 35


Ej. Codificación



- Detalle de la codificación de un Algoritmo Genético
- Sin perder generalidad podemos tratar sólo un problema de maximización

$$\min f(x) = \max g(x) = \max \{-f(x)\}$$

Algoritmos Genéticos 36




Ej. Codificación

- Asumimos que la función objetivo, de k variables, toma sólo valores positivos y está definida en un intervalo acotado

$$\max g(x) = \max\{f(x) + C\}$$

$$\text{Si } f(x_1, \dots, x_k) \geq 0 \forall x_i \wedge D_i = [a_i, b_i]$$

Algoritmos Genéticos 37




Ej. Codificación

- Si esperamos una precisión de seis dígitos decimales, se debe dividir el intervalo de búsqueda en mas de 1000000 secciones

$$\Rightarrow (b_i - a_i) \cdot 10^6$$

$$\Rightarrow (b_i - a_i) \cdot 10^6 \leq 2^{m_i} - 1$$

Algoritmos Genéticos 38




Ej. Codificación

- Se asigna un código a cada valor de x representado en el intervalo
- Aplicando la siguiente fórmula obtenemos el valor de x correspondiente con cada uno de los códigos

$$x = a_i + \text{decimal}(\text{string}_2) \frac{b_i - a_i}{2^{m_i} - 1}$$


Algoritmos Genéticos 39



Principio de funcionamiento

- Describiremos matemáticamente el principio de funcionamiento de los Algoritmos Genéticos
- Se basa en la representación en strings binarios de las posibles soluciones

Algoritmos Genéticos 40




Principio de funcionamiento

Definición de esquema

El esquema es una plantilla que permite encontrar similitudes entre los cromosomas

Algoritmos Genéticos 41




Principio de funcionamiento

- El Esquema (S) introduce el símbolo * en el alfabeto de los genes (0, 1)
- Ejemplo de un Esquema
 $S = *111100100$
- Coincide (Matching) con dos strings posibles
 $\{(0111100100), (1111100100)\}$

Algoritmos Genéticos 42


Principio de funcionamiento



- El símbolo * funciona como “comodín”
- Otro ejemplo
 $S = *1*1100100$
- Coincide con cuatro strings
 $\{(0101100100), (0111100100), (1101100100), (1111100100)\}$

Algoritmos Genéticos 43

Principio de funcionamiento




- Cada **esquema** coincide con 2^r **individuos**

Siendo r la cantidad de * del Esquema

Algoritmos Genéticos 44

Principio de funcionamiento



- Un **individuo** determinado de largo m coincide con 2^m **esquemas**.


Ej: 1001110001

Con

0 *	1001110001,
1 *	*001110001, 1*01110001, 10*1110001.....100111000*
2 *	**01110001, *1*1110001.....10011100**
.....
m-1 *	*****1, *****0*, *****0**.... 1*****
m *	*****

Algoritmos Genéticos 45

Principio de funcionamiento




- Un **individuo** determinado de largo m coincide con 2^m **esquemas**.

Recordar combinaciones $m! / r! (m - r)!$ y triángulo de Pascal

Con	
0 *	1
1 *	m
2 *	$m(m-1) / 2!$
.....	$m! / r! (m - r)!$
m-1 *	m
m *	1
Total =	2^m

Algoritmos Genéticos 46

Principio de funcionamiento




- Un **individuo** determinado, de largo m , coincide con 2^m **esquemas**.

Más fácil es considerar, simplemente, un sistema con sólo dos símbolos para cada bit del individuo, el erróneo o el que coincide.


Algoritmos Genéticos 47

Principio de funcionamiento



- Existen un total de 3^m esquemas posibles (0 1 *)
- En una población con n individuos se pueden representar entre 2^m y $n \cdot 2^m$ esquemas.


Algoritmos Genéticos 48



Propiedades del esquema

Orden del Esquema $o(S)$
 Se define como la cantidad de unos y ceros del Esquema
 Ej: $S = ***001*110$ $o(S) = 6$
 Define la especialización del Esquema
 Es útil para calcular la probabilidad de supervivencia del esquema frente a las mutaciones


Algoritmos Genéticos 49



Propiedades del esquema

Largo del Esquema $\delta(S)$
 Se define como la distancia entre la primera y la última posición del esquema con un 0 o un 1
 Ej: $S = ***001*110$ $\delta(S) = 10 - 4 = 6$
 Define si un esquema es compacto
 Es útil para calcular la probabilidad de supervivencia del esquema frente a las cruas


Algoritmos Genéticos 50



Propiedades del esquema

Aptitud del esquema en el tiempo t (o la generación G) :
 $eval(S,t)$
 Se define como el promedio de aptitud de todos los individuos (strings) en la población que coinciden con el esquema en el tiempo t
 $eval(S,t) = \sum_{j=1}^p \frac{eval(v_{ij})}{p} ; \{v_{ij}, \dots, v_{ip}\}$

Algoritmos Genéticos 51



Propiedades del esquema


Definimos la cantidad de strings que coinciden con el esquema S en el tiempo t como :

$$\xi(S,t)$$

$$\xi(S,t+1) = \xi(S,t) \cdot \frac{eval(S,t)}{F(t)} \cdot PopSize$$

Donde $F(t)$ es la suma de las adaptaciones individuales y $PopSize$ es la cantidad de individuos de la población.

Algoritmos Genéticos 52




Propiedades del esquema

- Sustituyendo por el promedio de aptitud de toda la población tenemos :

$$\overline{F(t)} = \frac{F(t)}{PopSize}$$

$$\xi(S,t+1) = \xi(S,t) \cdot \underbrace{\frac{eval(S,t)}{F(t)}}_{\text{Factor de crecimiento}}$$


Algoritmos Genéticos 53



Propiedades del esquema

- El número de strings en una población que coincide con un determinado esquema crece con una velocidad dada por la relación de la aptitud del esquema respecto al promedio de dicha población
- Se puede demostrar que para estos esquemas la cantidad de strings crece exponencialmente

Algoritmos Genéticos 54



Propiedades del esquema


Probabilidad de supervivencia del esquema ante la cruza

Dados dos esquemas:

$$S_0 = (*111*****) \quad \delta(S_0) = 2$$

$$S_1 = (11*****10) \quad \delta(S_1) = 9$$

Algoritmos Genéticos 55



Propiedades del esquema


La probabilidad de supervivencia P_s y la probabilidad de destrucción P_d ante la cruza

$$p_d(S) = \frac{\delta(S)}{m-1}$$

$$p_s(S) = 1 - \frac{\delta(S)}{m-1}$$

$$p_d(S_0) = \frac{2}{9} \quad p_d(S_1) = \frac{9}{9} = 1$$

Algoritmos Genéticos 56



Propiedades del esquema


Considerando ahora que los genes se cruzan con una probabilidad P_c , tenemos:

$$p_s(S) = 1 - p_c \frac{\delta(S)}{m-1}$$

Y la cantidad que se espera pase a la próxima generación:

$$\xi(S, t+1) = \xi(S, t) \cdot \frac{eval(S, t)}{F(t)} [1 - p_c \frac{\delta(S)}{m-1}]$$


Algoritmos Genéticos 57



Propiedades del esquema

- Significa que un esquema corto, de aptitud mayor al promedio, recibe un incremento exponencial de individuos en las generaciones futuras
- La cruce simple tiende a destruir esquemas largos y a combinar esquemas de bajo orden para construir esquemas de orden superior

Algoritmos Genéticos 58



Propiedades del esquema


Probabilidad de supervivencia ante la mutación

$$p_s(\text{bit}) = 1 - p_m$$

$$p_s(S) = (1 - p_m)^{o(S)}$$

Si $p_m \ll 1 \Rightarrow p_s(S) \cong 1 - o(S) \cdot p_m$

Algoritmos Genéticos 59




Propiedades del esquema

Y la cantidad que se espera pase a la próxima generación, sobreviviendo a la cruce y la mutación:

$$\xi(S, t+1) \cong \xi(S, t) \cdot \frac{\text{eval}(S, t)}{F(t)} \left[1 - p_c \frac{\delta(S)}{m-1} - o(S) \cdot p_m \right]$$

Recordar que se supuso a p_c y $p_m \ll 1$


Algoritmos Genéticos 60



Teorema del esquema

Esquemas cortos, de bajo orden y de aptitud superior al promedio de la población, reciben un incremento exponencial de individuos en las generaciones futuras !


Algoritmos Genéticos 61



Building blocks Hypothesis

Un algoritmo genético busca su performance sub-óptima a través de la superposición de esquemas cortos, de bajo orden y de aptitud elevada, llamados "Buildings Blocks"


Algoritmos Genéticos 62



Dos paralelismos...

- Exploración de varias soluciones a la vez.
- Posibilidad de implementación en paralelo.

Algoritmos Genéticos 63



Introducción a los
algoritmos genéticos

¡Gracias por la
atención!

Algoritmos Genéticos 64
