

UNIVERSIDAD DON BOSCO

Facultad De Ingeniería

Escuela De Computación

Sistemas Expertos e Inteligencia Artificial

Ciclo 02-2004



Algoritmos Genéticos

POR:

Ortega Martínez, Doménike Eunicee

Turcios Cubias, Boris Ernesto

CATEDRÁTICO:

Ing. Cruz Galdamez

Ciudadela Don Bosco, Miércoles 23 de Junio de 2004

Algoritmos Genéticos

Concepto.

Medir la inteligencia es averiguar el empleo de todo lo que se sabe en forma útil. Se trata de hallar soluciones a problemas ya sea consultando soluciones previas ("preparación") o combinando posibles soluciones ("deliberación"). Hay infinitos métodos de búsqueda. El más trivial y demorado es el de búsqueda del óptimo por fuerza bruta (revisar sistemáticamente todo el espacio de problema) y los más sofisticados, más breves, apelan a matemáticas que pueden ser tan complicadas como el autor quiera - sujeto a que su lógica sea entendible por algún experto.

En bioingeniería se imita la forma como la naturaleza ha resuelto estos temas. Se trata de combinar, en los mecanismos evolutivos, las mecánicas de la selección natural con las de la genética natural ("algoritmos genéticos"), que sirven de inspiración para los estudios de redes neuronales cada vez más aptas para el logro de metas.

Para poder comprender mejor el significado, se describe la definición individual de cada uno de los términos que lo componen:

Algoritmo.

Método repetitivo para resolver problemas. Secuencia codificada de instrucciones para manipulación de símbolos. Un algoritmo genera un "*proceso algorítmico*", diseñado intencionalmente o no (motivo este último por el cual se hace la hipótesis que la selección natural o la actividad intelectual son procesos algorítmicos). En general, el proceso consiste en la obediencia a una estructura única, ramificada, recursiva o iterativa, que se va desarrollando en serie-paralelo, con rutinas y subrutinas que se llaman al ser necesarias. Se discute si realmente el proceso intelectual de *comprensión de la realidad* es un proceso algorítmico o lo

supera (postura, por ejemplo, de Roger Penrose). A veces se contraponen los algoritmos a las redes neurales artificiales, pese a que el aprendizaje de estas últimas se realiza con algoritmos.

Genética.

El estudio de la herencia en seres vivos. Los genes llevan consigo la información codificada en la molécula replicante (con la ayuda de enzimas) de DNA. Esa información determina la fisiología de la especie en el proceso del desarrollo u ontogenia, con la ayuda - en seres vivos avanzados - del efecto materno. Un mensaje genético adecuado tiende a producir organismos capaces de sobrevivir al ambiente mientras que un mensaje genético inadecuado tiende a generar una supervivencia difícil.

Como definición formal se puede decir que el Algoritmo Genético "es un algoritmo de búsqueda basado en la mecánica de la selección natural y de la genética natural. Combina la supervivencia del más apto entre estructuras de secuencias con un intercambio de información estructurado, aunque aleatorizado, para constituir así un algoritmo de búsqueda que tenga algo de las genialidades de las búsquedas humanas" [Goldberg, 1989].

El algoritmo genético permite obtener soluciones a un problema que no tiene ningún método de resolución descrito de forma precisa, o cuya solución exacta, si es conocida, es demasiado complicada para ser calculada en un tiempo aceptable. Es el caso particular de cuando se encuentran restricciones múltiples y complejas e incluso contradictorias que deben ser satisfechas simultáneamente como por ejemplo para formar equipos de trabajo, planificar rondas de entregas, implantar puntos de venta de manera óptima, construir modelos estadísticos.

Limitaciones

- ⊕ La codificación del genoma es fundamental en un problema de algoritmos genéticos. Veamos con más detalle toda la problemática que hemos encontrado en la implementación de nuestro algoritmo.
- ⊕ Los algoritmos genéticos son muy robustos para casi todos los problemas definidos en conjuntos discretos, pero en conjuntos continuos su comportamiento ha de ser analizado *a priori*. Hemos de buscar una codificación de los datos en el genoma para la cual tengamos un operador de cruce tal que el grado de adaptación del descendiente sea función del grado de adaptación de los padres. En los problemas continuos esto puede no ser fácil de localizar, hasta puede no existir. Habitualmente va a estar en función del grado de *suavidad* de la función de adaptación. En una función de adaptación poco suave, será más difícil encontrar esta codificación.
- ⊕ En la resolución de un mismo problema el enfoque algorítmico es específico, muy rápido mientras el algoritmo genético se caracteriza por ser general pero muy lento.

Tecnología

Existen varios paquetes y bibliotecas de algoritmos genéticos en el mercado, muchos de ellos muy buenos. Sin embargo, las características tan específicas de nuestro problema nos hicieron al final decidirnos por reimplementar todo el algoritmo nosotros, en vez de emplear algún paquete preexistente. Sin embargo, existen paquetes de mucha calidad; algunos de ellos son:

- ✦ **GALOPPS**: Bastante bueno, es muy flexible y además nos permite con facilidad puede ser encontrado en *GARAGe*. Su dirección primaria en Internet es GARAGe.cps.msu.edu/software/software-index.html, y su dirección para descargarlo vía FTP es [arage.cps.msu.edu/pub/GA/galopps/](ftp://arage.cps.msu.edu/pub/GA/galopps/)

- ✦ **GAGS**: Generador de aplicaciones basadas en algoritmos genéticos, escrito en C++. Desarrollado por el grupo de J.J. Melero. Excelente. Su dirección Web es kal-el.ugr.es/gags.html, y su dirección para descargarlo vía FTP es [kal-el.ugr.es/GAGS/](ftp://kal-el.ugr.es/GAGS/).

- ✦ **FORTRAN GA**: Desarrollo de algoritmos genéticos para Fortran. Su dirección Web es www.staff.uiuc.edu/~carroll/ga.html.

- ✦ **Galib**: Biblioteca de algoritmos genéticos de Matthew. Conjunto de clases en C++ de algoritmos genéticos. Su dirección Web es lancet.mit.edu/ga/, y su dirección para descargarlo vía FTP es [lancet.mit.edu/pub/ga/](ftp://lancet.mit.edu/pub/ga/). Podemos registrarlo en <http://lancet.mit.edu/ga/Register.html>.

- ✦ **GAS:** Paquete para desarrollar aplicaciones de algoritmos genéticos en Python. Su dirección Web es starship.skyport.net/crew/gandalf, y su dirección para descargarlo vía FTP es [ftp.coe.uga.edu/users/jae/ai](ftp://ftp.coe.uga.edu/users/jae/ai).

- ✦ **GECO:** Conjunto de herramientas para Lisp. Su dirección para descargarlo vía FTP es <ftp://ftp.aic.nrl.navy.mil/pub/galist/src/>.

- ✦ **GPdata:** Para desarrollar algoritmos genéticos en C++. Su dirección para descargarlo vía FTP es [ftp.cs.bham.ac.uk/pub/authors/W.B.Langdon/gp-code/](ftp://ftp.cs.bham.ac.uk/pub/authors/W.B.Langdon/gp-code/), y su documentación -GPdata-icga-95.ps- la podemos encontrar en el *site* de Internet cs.ucl.ac.uk/genetic/papers/.

- ✦ **gpjpp:** Bibliotecas de clases para desarrollar algoritmos genéticos en Java Su dirección Web es [www.turbopower.com/~ kimk/gpjpp.asp](http://www.turbopower.com/~kimk/gpjpp.asp).

- ✦ **GP Kernel:** Biblioteca de clases para programación genética en C++. Su dirección Web es [www.emk.e-technik.th-darmstadt.de/~ thomasw/gp.html](http://www.emk.e-technik.th-darmstadt.de/~thomasw/gp.html).

- ✦ **lil-gp:** Herramientas para programación genética en C. Su dirección Web es isl.msu.edu/GA/software/lil-gp/index.html, y su dirección para descargarlo vía FTP es [isl.cps.msu.edu/pub/GA/lilgp/](ftp://isl.cps.msu.edu/pub/GA/lilgp/). Podemos encontrar los parches para Linux en www.cs.umd.edu/users/seanl/patched-gp.

- ✦ **PGAPack:** *Parallel Genetic Algorithm Library*. Biblioteca de algoritmos genéticos paralelos. Podemos encontrarlo en la dirección de Internet con un navegador en www.mcs.anl.gov/home/levine/PGAPACK/index.html, y su dirección para descargarlo vía FTP es [ftp.mcs.anl.gov/pub/pgapack/](ftp://mcs.anl.gov/pub/pgapack/).

 - ✦ **Sugal:** *SUnderland Genetic ALgorithm system*. Para hacer experimentos con algoritmos genéticos. Podemos encontrarlo en la dirección de Internet con el navegador en www.trajan-software.demon.co.uk/sugal.htm.

 - ✦ **ADATE:** *Automatic Design of Algorithms Through Evolution*. Programación evolutiva. Su dirección Web es www-ia.hiof.no/~rolando/adate_intro.html.

 - ✦ **GPsys:** Sistema de programación genética en Java. Podemos encontrarlo en la dirección de Internet www.cs.ucl.ac.uk/staff/A.Qureshi/gpsys.html.
-

Aplicaciones

- ✦ El algoritmo se muestra especialmente útil cuando la función no es "suave" desde el punto de vista analítico, es decir, que no es derivable o es complicada de derivar, incluso puede no ser continua. En esta situación las técnicas del análisis no se pueden aplicar y se necesita otro método para

calcular los óptimos, es en este caso cuando el algoritmo genético entra en juego, aunque está claro que también se puede usar en los otros casos.

La interfase entre la ingeniería del conocimiento y la tecnología de la educación basada en modelos bioingenieriles, es una perspectiva abierta: lo que realmente el ser humano debe saber es qué mecanismos usa (y debe rediseñar cuidadosamente, si fuera posible) para elaborar su pensamiento, aprovechar las emociones útiles, dominar las emociones antisociales y tomar decisiones.

Según el algoritmo genético, numerosas soluciones más o menos correctas inherentes a dicho problema son creadas al azar, según una forma ya definida : itinerario, horarios, base de reglas de decisión, evaluación por puntuación, red neuronal, etc.

- ⊕ El algoritmo genético puede aplicarse a la producción de una variedad de objetos mientras sea posible obtener una calificación que represente la justeza de la solución.

- ⊕ Es posible fabricar previsores estadísticos no a través de cálculos de datos como en la estadística clásica sino haciendo evolucionar los datos por algoritmo genético ("inducción"). Por problemas de clasificación o de segmentación, la justeza significa simple y llanamente la tasa de reordenación del previsor con respecto a un conjunto dado de ejemplos. El mecanismo de estimulación de lo más apto permite entonces la aparición del previsor que reordenará los datos lo mejor posible. Este tipo de construcción de previsor forma parte de las llamadas técnicas de [data mining](#).

Los previsores producidos pueden tener formas muy diversas : bases de reglas, evaluación por puntuación, árboles de decisión e incluso redes neuronales.

Información General

Antecedentes

En 1859, Darwin publica su libro «*El origen de las especies*». Este libro causó una agria polémica en el mundo científico por las revolucionarias teorías en él contenidas: que las especies evolucionan acorde al medio, para adaptarse a éste. Con ello, el universo pasaba de ser una creación de Dios, estática y perfecta desde su inicio, a un conjunto de individuos en constante competición y evolución para poder perpetuar su especie en el tiempo. La existencia de una especie pasa así a ser algo dinámico; las especies se crean, evolucionan y desaparecen si no se adaptan. Para cada especie animal, la naturaleza proponía un crudo filtro: sólo los mejores, los más aptos, los que mejor se adaptan al medio conseguían sobrevivir lo suficiente para llegar a la madurez, y encontrar una pareja para perpetuar sus aptitudes que le hacían más apto.

La informática ve aquí un claro proceso de optimización. Tomamos los individuos mejor adaptados -mejores soluciones temporales-, los cruzamos -mezclamos-, generando nuevos individuos -nuevas soluciones- que contendrán parte del código genético -información- de sus dos antecesores, y, por lo tanto, aunque el nuevo individuo no tenga que estar forzosamente mejor adaptado --de hecho, puede que ni la probabilidad de que el nuevo individuo generado esté mejor adaptado que los padres sea alta-, el promedio de la adaptación de toda la población si mejora, ya que tienden a perpetuarse y extenderse las mejores características, y a extinguirse las poco beneficiosas o perjudiciales. Aquí vemos que, a diferencia de los métodos anteriormente citados, no tenemos por qué desarrollar un individuo mejor. En los algoritmos genéticos creamos una nueva abstracción -la población- cuya función de coste mejorará globalmente, por lo que puede que nos encontremos algún individuo con mejores características.

Ironías de la historia; en 1833 el gran amigo personal de Darwin, Charles Babbage, diseñaba su máquina analítica.

John Holland en 1975 en su célebre artículo [Hol75], intuyó la posibilidad de incorporar la semántica de la evolución natural a procesos de optimización, y comenzaron así los estudios en algoritmos genéticos.

Descripción de Funcionamiento

El algoritmo genético que se basa en la adaptación de los individuos al medio en el que viven, podemos hacer una analogía de este algoritmo con la evolución de una especie en un ecosistema.

Damos algunas definiciones:

- ✦ *Individuo* es un punto dentro de ese intervalo.
- ✦ *Genotipo*, las características de cada individuo, es un punto en el intervalo en el que queremos calcular el mínimo.
- ✦ *Adaptación* de un individuo al medio es la proximidad de la imagen de ese punto por la función al extremo mínimo.
- ✦ *Población inicial* es el conjunto de puntos en el momento en el que se inicia el algoritmo.

⊕ *Hijo de dos individuos* es un punto que está entre ambos puntos padres (de aquí el nombre de genético, se toman los individuos mejor adaptados, con menores valores imagen, y se crean nuevos puntos que estén entre esos dos puntos, se puede decir que el hijo hereda el código genético de los dos padre y por tanto sus propiedades de adaptación al medio.) Para crear nuevos hijos se toman los individuos mejor adaptados.

⊕ *Número máximo de la población* es el número de individuos que soporta ese medio, el punto fijo de la ecuación logística para ese ecosistema.

⊕ *Muerte*, desaparecen del conjunto de puntos algunos puntos escogidos al azar, dando preferencia a los que están menos adaptados. Se quitan tantos puntos como hagan falta hasta llegar al punto fijo de la población.

⊕ *Mutación*, cada cierto número de nacimientos se producen unas mutaciones, que son individuos nuevos que se añaden a la población con código genético cambiado, sirven para que la especie no se estanquen en un mínimo local.

Se utilizan probabilidades para que se mueran los individuos menos adaptados y se reproduzcan los más adaptados, probabilidades que favorecen la reproducción de los individuos mejor adaptados al medio y la muerte de los peores adaptados.

Después de algunas iteraciones se ve como la población tiende a un mínimo, podemos decir que la especie se especializa en ese ecosistema pero puede que

esa adaptación esa en un mínimo local de la función y que la especie se estanque en él y no sea capaz de tomar valores más bajos, por eso se incorporan mutaciones para que la población salte a otro mínimo (quizás de menor valor) y que se produzcan individuos mejor adaptados, este proceso termina cuando la población alcanza el mínimo global en el intervalo dado pues por muchas mutaciones que produzcan después morirán al no estar tan bien adaptadas como la población general que ya se ha adaptado muy bien al medio (aprovecha todos los recursos disponibles en el medio).

Estudiando el algoritmo se van muchas similitudes con especies de la vida real, aparte de las ya explicadas podemos ver.

Si el número de la población inicial es muy grande la especie llega antes al óptimo de su adaptación al medio, al igual que pasa en la vida real cuando hay mucha biodiversidad siempre aparecen especies muy bien adaptadas al medio. Si hay pocos individuos inicialmente es muy probable que se metan en un mínimo local (los individuos de esa especie serán muy parecidos entre sí y no aprovecharán otros recursos disponibles en el medio) y si no hay suficientes mutaciones le cuesta salir de él. Si hay suficientes mutaciones, con el tiempo, se creará una nueva especie que aprovechará más recursos del medio y dominará el ecosistema haciendo que la especie menos adaptada se extinga.

Hay otros factores que influyen en gran medida, como es el número de hijos por cada dos individuos, las veces que se reproducen, el número de la población, todos ellos tienen la propiedad de que cuanto mayor sean más probable es que se alcance el óptimo, claro que también es más pesado calcular una iteración pues hay más individuos que tratar. Otro factor importante es la muerte que es lo que produce que desaparezcan las especies peor adaptadas.

Características de los algoritmos genéticos

Los algoritmos genéticos presentan ventajas sobre los otros algoritmos comentados en el capítulo anterior que los hacen preferibles para determinado tipo de aplicaciones, entre las cuales está la optimización de agregados del Silicio, que corresponde a nuestro problema. Algunas de las características de los algoritmos genéticos son:

- ⊕ **Son algoritmos estocásticos.** Dos ejecuciones distintas pueden dar dos soluciones distintas. Esto es útil por el hecho de que hay gran cantidad de isómeros que corresponden a soluciones válidas, por lo que es interesante que distintas ejecuciones nos puedan dar isómeros distintos.
- ⊕ **Son algoritmos de búsqueda múltiple,** luego dan varias soluciones. Aunque habitualmente las energías de los individuos de la población final es similar, los individuos suelen ser distintos entre sí. Con el modelo de paralelización empleado -genético multipoblacional- la probabilidad de obtener muchas soluciones distintas es más alta todavía. Por ello, nos podemos quedar con la solución que más nos convenga según la naturaleza del problema; en nuestro caso, analizar los isómeros generados por el algoritmos para decidir cuáles son más interesantes para su posterior estudio físico. Por lo comentado en el punto anterior, esto supone una ventaja en nuestro problema.
- ⊕ **De los algoritmos comentados en el capítulo anterior, son los algoritmos que hacen una barrida mayor al subespacio de posibles soluciones válidas, y con diferencia.** De hecho, se considera que, de todos los algoritmos de optimización estocásticos, los algoritmos genéticos son de los más exploratorios disponibles.
- ⊕ A diferencia de los otros algoritmos comentados, cuya convergencia y resultado final son fuertemente dependientes de la posición inicial, en los algoritmos genéticos -salvo poblaciones iniciales realmente degeneradas,

en los que el operador de mutación va a tener mucho trabajo- **la convergencia del algoritmo es poco sensible a la población inicial** si esta se escoge de forma aleatoria y es lo suficientemente grande.

- ✦ Por su grado de penetración casi nulo, la curva de convergencia asociada al algoritmo presenta una **convergencia excepcionalmente rápida al principio**, que casi enseguida se bloquea. Esto se debe a que el algoritmo genético es excelente descartando subespacios realmente malos. Cada cierto tiempo, la población vuelve dar el *salto evolutivo*, y se produce un incremento en la velocidad de convergencia excepcional. La razón de esto es que algunas veces aparece una mutación altamente beneficiosa, o un individuo excepcional, que propaga algún conjunto de cromosomas excepcional al resto de la población. La pobre penetración es el agujero más importante en los algoritmos genéticos, y la razón fundamental de la aparición de los algoritmos híbridos y miméticos, de los que hablaremos en el próximo capítulo. De hecho, no hemos empleado en las últimas versiones un genético puro; sino que comenzamos hibridizando el algoritmo genético con una agregación simulada que nos permitía desbloquear la convergencia cuando esta quedaba bloqueada, con lo cual acelerábamos los saltos evolutivos. Finalmente hemos acabado recurriendo a un algoritmo mimético, en el cual en lugar de emplear la agregación simulada para generar saltos evolutivos empleamos el pseudogradiante para que todos los elementos de la población sean los mínimos locales de sus subespacios. Con ello, compiten entre sí los mínimos locales de los subespacios, por lo que la velocidad de los saltos evolutivos se acelera bastante, además de incrementarse la velocidad de convergencia.
- ✦ **La optimización es función de la representación de los datos.** Este es el concepto clave dentro de los algoritmos genéticos, ya que una buena codificación puede hacer la programación y la resolución muy sencillas, mientras que una codificación errada nos va a obligar a estudiar que el nuevo genoma cumple las restricciones del problema, y en muchos problemas tendremos que *abortar* los que no cumplan las restricciones, por

ser estas demasiado complejas. Además, la velocidad de convergencia va a estar fuertemente influenciada por la representación. Nuestra representación en nuestro problema de optimizar agregados del Silicio es independiente de la traslación del agregado -nosotros hemos normalizado la codificación respecto al centro de masas-, pero dependiente de la rotación. Además, el operador de cruce ha presentado una problemática muy interesante que comentaremos en una sección posterior.

✦ **Es una búsqueda paraméricamente robusta.** Eso quiere decir que hemos de escoger realmente mal los parámetros del algoritmo para que no converja. Con tasas razonables, va a converger -mejor o peor- en una solución razonablemente buena si la representación es la adecuada. Esto es muy importante por la naturaleza de nuestra búsqueda. Nosotros no podemos hacer comparativas buscando los mejores *números mágicos* para que nuestro algoritmo converja, ya que el objetivo de nuestro grupo de investigación es llegar a la solución de un problema que ya de por sí es muy complejo. Por ello necesitamos un algoritmo en el cual podamos equivocarnos en los parámetros de partida, y estos son los algoritmos genéticos.

✦ Por último, **los algoritmos genéticos son intrínsecamente paralelos.** Esto significa que, independientemente de que lo hayamos implementado de forma paralela o no, buscan en distintos puntos del espacio de soluciones de forma paralela. Ese paralelismo intrínseco permite que sean fácilmente paralelizables, es decir, que sea fácil modificar el código para que se ejecute simultáneamente en varios procesadores. Esto nos ha ayudado a desarrollar rápidamente una solución paralela, que comentaremos en un capítulo posterior.

Decisiones para implementar un algoritmo genético

Las decisiones que hay que tomar para implementar un algoritmo genético son:

- *Criterio de codificación.* Como se va a almacenar la información en el genoma. En nuestra aplicación, el criterio de codificación será la secuencia de las coordenadas del agregado después de haber desplazado el centro del eje de coordenadas al centro de masas.
- *Criterio de tratamiento de individuos no factibles.* Como se van a tratar a los individuos que no cumplan las restricciones.
- *Criterio de inicialización.* Cómo se va a construir la población inicial del algoritmo genético.
- *Criterio de parada.* Determina cuándo el algoritmo ha llegado a una solución aceptable.
- *Función de adaptación.* Corresponde a la función de costo de la investigación operativa tradicional.
- *Operadores genéticos.* Se emplean para determinar cómo va a ser la nueva generación. Básicamente son los operadores de cruce y mutación, aunque pueden ser empleados otros adicionales -muerte, aborto,

envejecimiento...-. Tanto cruce como mutación pueden ser realizados de muchas formas distintas.

- *Criterios de reemplazo.* Los criterios que determinan quiénes se van a cruzar. No tienen que ser obligatoriamente los mismos que los criterios de selección de los padres.

 - *Parámetros de funcionamiento.* Determinados parámetros que, sin poder ser englobados en ninguno de los anteriores, son fundamentales para el funcionamiento de un algoritmo genético. Es el caso, por ejemplo, del tamaño de la población, la probabilidad de la aplicación de los operadores genéticos.
-