

Introducción a los algoritmos genéticos

Introduction to genetic algorithms introduction à algorithmes génétiques

Recepción: 31-05-2010
Evaluación: 20-07-2010
Aceptación: 30-07-2010
Artículo de Revisión

* Jairo Amador Niño

Resumen

Este documento presenta una introducción a los algoritmos genéticos AGs, como resultado de la revisión bibliografía por parte del Grupo de Investigación de Inteligencia Artificial de la Universidad Santo Tomás, haciendo énfasis en los siguientes aspectos: los AGs, la robustez en la optimización tradicional y en los métodos de búsqueda, sus metas, algunos campos de aplicación, y como funcionan.

Palabras clave: Optimización, algoritmos genéticos y robustez.

Abstract

This paper presents an introduction to genetic algorithms GAs as a result of the literature review done by the research group on Artificial Intelligence at Santo Tomás University, emphasizing the following aspects: GAs, the robustness in the traditional optimization and search methods, their goals, areas of application, and how they work.

Key words: Optimization, genetic algorithms, robustness.

Résumé

Ce document présente une aux algorithmes génétiques AGs, comme résultat de la révision bibliographie de la part du Groupe de Recherche d'Artificielle Intelligence de l'Université Saint Tomas, en faisant une emphase dans les aspects (apparences) suivants: les AGs, la robustesse dans l'optimisation traditionnelle et dans les méthodes de recherche, ses gardiens de but, quelques champs d'application, et comme ils(elles) fonctionnent.

Mots-clés: Optimisation, algorithmes génétiques et de robustesse

Qué son los algoritmos Genéticos

Los algoritmos genéticos son algoritmos de búsqueda, basados sobre mecanismos de selección y genética natural concordante con la teoría de la evolución de Charles Darwin. Ellos favorecen la supervivencia de los más capaces de un conjunto de cadenas estructuradas en bits generalmente, que intercambian información de manera pseudoaleatoria manteniendo la estructura de la cadena, para poder aplicar un algoritmo de búsqueda con algo de intuición innovadora por parte del ser

humano. En cada generación, un nuevo conjunto de individuos artificiales o cadenas estructuradas son creadas, utilizando bits y fragmentos de las mismas; cadenas que han sido favorecidas de alguna manera por la asignación de una medida. Los algoritmos genéticos no son simples pasos del azar, ellos explotan eficientemente la información histórica para explorar sobre nuevos puntos de búsqueda con el propósito de encontrar mejores soluciones.

* Magíster en Ciencias Computacionales. Docente Departamento de Ciencias Básicas, Universidad Santo Tomás, Seccional Tunja. Grupo de Investigación CB-USTAT. jairoamadornino@hotmail.com

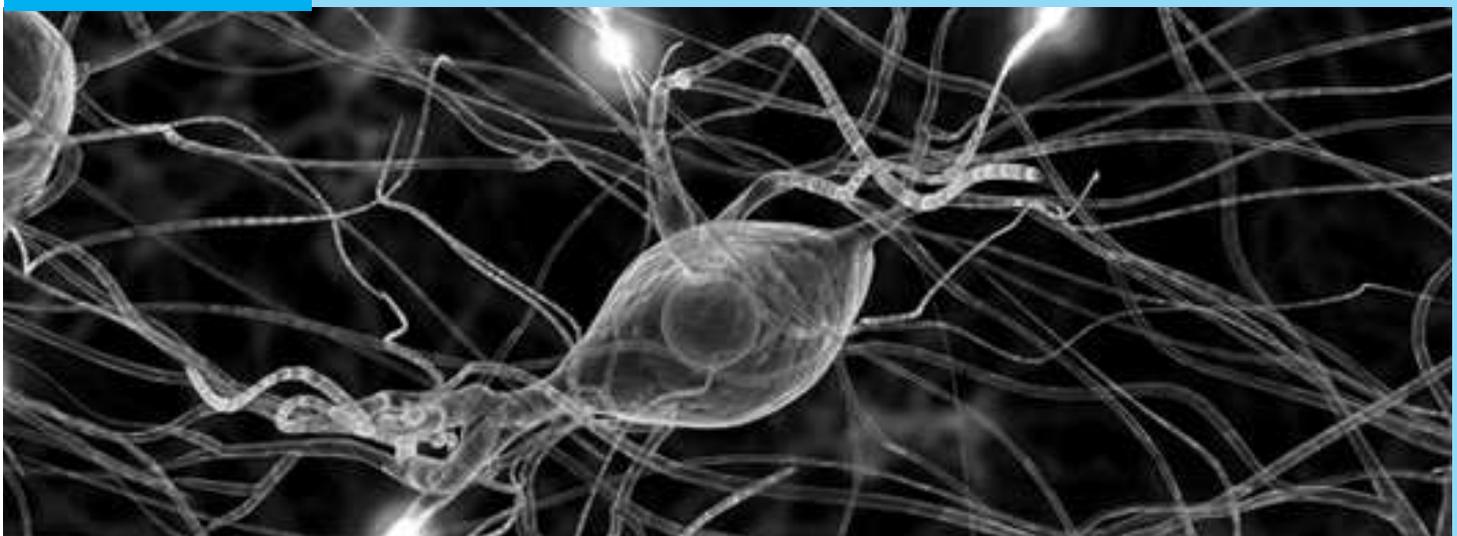


Los AGs han sido desarrollados por John Holland, sus colegas y sus estudiantes en la Universidad de Michigan. Las metas de la investigación eran dos: 1) Abstractar y explicar rigurosamente los procesos adaptativos de los sistemas naturales, y 2) Haciendo uso de software diseñar sistemas artificiales que retuvieran los mecanismos más importantes de los sistemas naturales. Esto lo llevó a importantes descubrimientos de la ciencia en los sistemas naturales y artificiales.

El tema central del estudio de los algoritmos genéticos ha sido su robustez, el balance entre la eficiencia y la eficacia necesaria para que las cadenas estructuradas sobrevivan en diferentes entornos. Las implicaciones de la robustez para los sistemas artificiales son varias, si estos pueden hacerse más robustos, los rediseños costosos son ampliamente reducidos, si los niveles más altos de adaptación pueden ser alcanzados, estos mismos sistemas pueden ejecutar sus funciones en forma más prolongada y mejor. Los diseñadores de sistemas artificiales como software, hardware, computación, negocios e ingeniería, solo pueden maravillarse de la robustez, eficiencia y la flexibilidad de estos sistemas “que no tienen qué envidiarle” a los sistemas biológicos, caracterizados por su autorecuperación, autoadministración y reproducción, reglas propias de los sistemas biológicos, y que escasamente existen en los sistemas artificiales más sofisticados.

Así se llega a una interesante conclusión, donde el desempeño robusto es deseado (y dónde no?) sabiendo que la naturaleza es la que lo hace mejor; los secretos de adaptación y supervivencia son mejor aprendidos del estudio cuidadoso de los ejemplos biológicos. Los AGs no son solo aceptados por el único argumento de la belleza natural. Los AGs teórica y empíricamente han sido probados y proporcionan búsquedas robustas en espacios complejos. La monografía más importante sobre este tema es el trabajo de Holland (1975) titulado “Adaptation in natural and artificial systems”. Muchos artículos y disertaciones establecen la validez de la técnica en optimización y aplicaciones de control. Se ha establecido como un método válido para problemas que requieren búsquedas eficientes y efectivas. Los AGs han encontrado grandes aplicaciones en los campos científicos, de negocios e ingeniería. La razón detrás de esto es el número creciente y claro de aplicaciones. Estos algoritmos son computacionalmente sencillos y poderosos en procesos de búsqueda. Además los AGs no están limitados por supuestas restricciones del espacio de búsqueda, tales como tener continuidad, existencia de derivadas, unimodalidad y otras condiciones. Se investigan las razones que hay en estas cualidades atractivas; pero antes de esto, se explota la robustez en los procedimientos de búsqueda que han sido ampliamente aceptados.

Fuente: <http://www.revistacronopio.com/wp-content/uploads/2009/12/neurona02.jpg>



La robustez en la optimización tradicional y en los métodos de búsqueda

No se pretende hacer un estudio comparativo de técnicas de búsqueda y optimización. No obstante, es importante preguntarse si los métodos de búsqueda convencionales tienen los requerimientos de robustez. La literatura actual identifica tres tipos principales como métodos de búsqueda: basados en el cálculo, enumerativos y aleatorios. Se comentará cada tipo para ver qué conclusiones se pueden hacer sin una prueba formal.

Los métodos basados en el cálculo han sido estudiados ampliamente; estos se subdividen en dos clases principales: indirecta y directa. Los métodos indirectos buscan el extremo local para resolver el conjunto de ecuaciones usualmente no lineales resultantes de la configuración del gradiente de la función objetivo igualada a cero. Esta es la generalización multidimensional de la noción de cálculo elemental para hallar puntos extremos, máximos y mínimos, como se ilustra en la figura 1.1

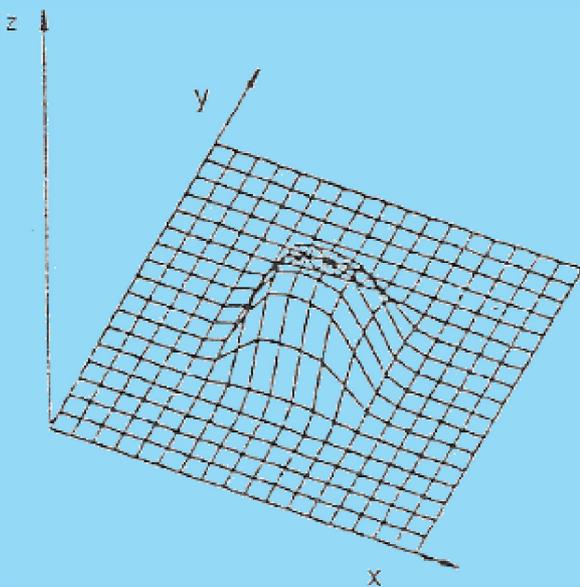
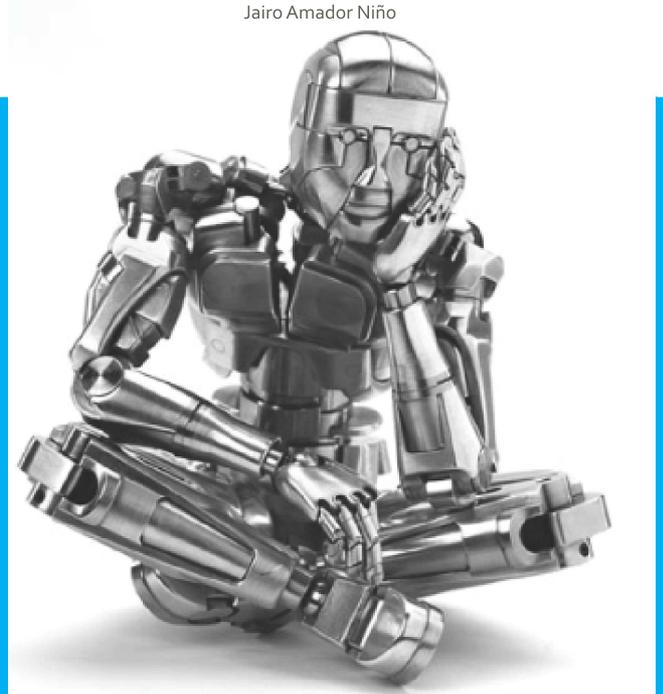


Figura 1.1 La función tiene solo un máximo y es fácil obtenerlo mediante cálculo.



Fuente:
http://1.bp.blogspot.com/_kVAepF32L84/TA7zo2Qill/AAAAAAAAAAc/PCZQx7-AdHo/s1600/zoho.jpg

Dada una función no restringida y suave, encontrar un posible pico que comience con búsquedas restringidas para estos puntos con pendiente cero en todas las direcciones. De otra manera los métodos de búsqueda directa buscan el óptimo local a intervalos o saltos sobre la función, moviéndose en dirección hacia el gradiente local. Esto es simplemente la noción de la escalada de la colina para encontrar el máximo local escalando la función en dirección a la posible parte más empinada. Mientras que estos dos métodos basados en el cálculo han sido mejorados, ampliados, detallados y difundidos, algunos razonamientos simples muestran su falta de robustez.

Primero, ambos métodos son locales en su alcance: el óptimo que ellos buscan son el mejor en la vecindad de un determinado punto. Por ejemplo suponga que la figura 1.1 muestra una porción del dominio completo de interés, un gráfico más complejo es mostrado en la figura 1.2 Claramente podría presentarse que comenzando la búsqueda, o cuando los procedimientos encuentran un cero en la vecindad del pico más bajo, se producirá la pérdida del evento principal (el pico más alto). Además, una vez el pico más bajo haya sido alcanzado, mejoras adicionales deben ser investigadas y tomadas a través de un reinicio aleatorio u otro procedimiento.

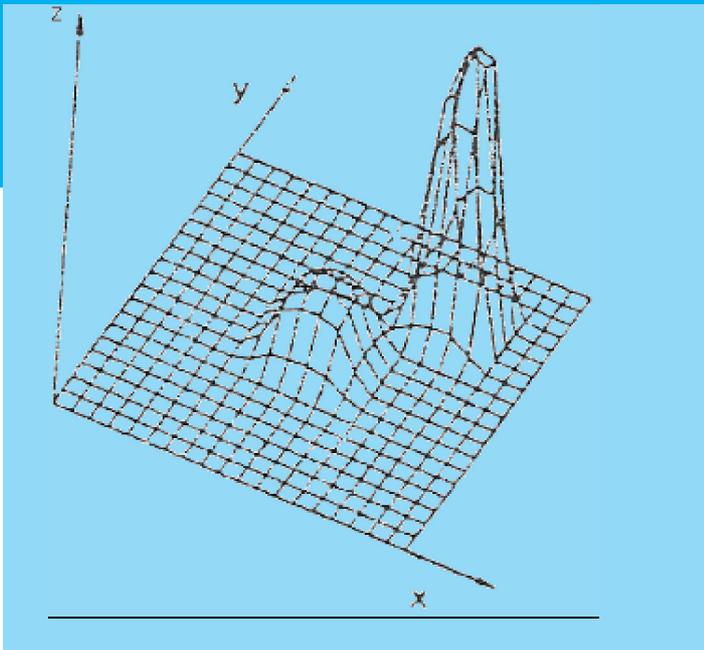


Figura 1.2 La función con múltiples picos genera un dilema, cual colina debe ser escalada?

Segundo, los métodos basados en el cálculo dependen de la existencia de las derivadas (valores de la pendiente bien definidos). Aún, si se permiten aproximaciones numéricas de las derivadas, esto es un serio defecto. Muchos parámetros prácticos del espacio se quedan cortos respecto a la noción de derivada y a la suavidad de la curva que esto implica. Los teóricos interesados en optimización han sido también complacientes en aceptar el legado de los matemáticos de los siglos dieciocho y diecinueve, quienes pintaron un mundo con restricciones lineales, y que en el presente heredamos. El mundo real de búsquedas está cargado de discontinuidades, grandes multimodales, espacios de búsqueda ruidosos, tal como se describe en algunas funciones del cálculo, como se muestra en la figura 1.3.

No es sorprendente que los métodos de solución de algunos problemas dependan de requerimientos como la continuidad y la existencia de la derivada, y solo se aplique en un dominio limitado del problema o de problemas. Por esta razón, y por su inherente alcance local de búsqueda, se hace importante la aplicación de los AGs en la solución de algunos problemas, y esto muestra que algunas veces el cálculo no es lo suficientemente robusto en dominios imprevistos.

Hablando de los esquemas enumerativos, estos han sido considerados de muchas formas y tamaños. La idea es sumamente directa, en

espacios de búsqueda finita o infinita discreta. Aunque la simplicidad de este tipo de algoritmo es atractivo, y la enumeración es una clase de búsqueda muy humana cuando el número de posibilidades es pequeño, tales esquemas deben ser finalmente reducidos en la competencia de la robustez por una simple razón: la pérdida de eficiencia.

Los algoritmos de búsqueda aleatoria han alcanzado popularidad creciente en la medida que los investigadores han reconocido las limitaciones de los esquemas basados en cálculos y en la enumeración. Aunque, el comportamiento y los esquemas aleatorios que buscan y guardan el mejor, deben también ser descontinuados por el requerimiento de eficiencia. En las búsquedas aleatorias en corridas de tiempo largo puede esperarse que no se logre el mejor esquema enumerativo. En la prisa por reducir estrictamente los métodos de búsqueda aleatoria, se debe ser cuidadoso para separar estos de las técnicas aleatorias.

El AG es un ejemplo de un procedimiento de búsqueda, que utiliza selección aleatoria como una herramienta para lograr una búsqueda con alta explotación, sobre la codificación del espacio parametrizado. Utilizar selección aleatoria como una herramienta en un proceso de búsqueda directa es algo extraño en principio. Pero la naturaleza contiene algunos ejemplos. Otra técnica de búsqueda

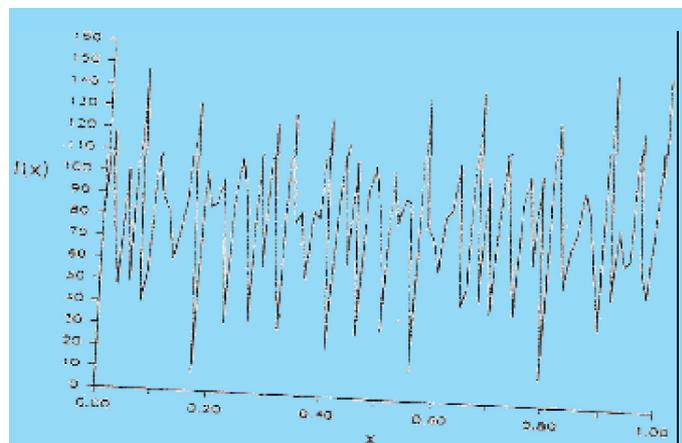


Figura 1.3 Algunas funciones son discontinuas, ruidosas e inapropiadas para los métodos tradicionales de búsqueda.

actualmente popular es el recocido simulado, que usa procesos aleatorios para ayudarse en el encaminamiento de su propia forma de buscar, para minimizar la energía gastada en la búsqueda. El libro (Davis, 1987) explora las relaciones entre el recocido simulado y los algoritmos genéticos. Algo importante, es reconocer en este momento crítico, que las búsquedas aleatorias no necesariamente implican búsquedas menos directas.

Esta discusión no ha sido un examen exhaustivo de los métodos vistos en optimización tradicional. Quedan cosas pendientes que pueden alterar la siguiente conclusión: los métodos convencionales de búsqueda no son robustos, esto no implica que los métodos no sean útiles. Los esquemas mencionados e incontables combinaciones híbridas y las permutaciones han sido utilizadas exitosamente en muchas aplicaciones, sin embargo, como más problemas complejos son atacados, serán necesarios otros métodos. Al inspeccionar el espectro del problema de la figura 1.4, la técnica de gradiente es bien ejecutada en la parte angosta de esta clase de problemas, como se esperaba, pero esto llega a ser altamente ineficiente en otro caso. De otra manera, el esquema enumerativo se ejecuta con igual ineficiencia a través del espectro de problemas como se ve en la curva de ejecución en la parte inferior.

Podría ser que valga la pena sacrificar la ejecución máxima en un problema en particular para lograr un relativo nivel alto de ejecución sobre el espectro de problemas. A propósito, con varios métodos eficientes se podrían crear esquemas híbridos que ayuden a completar la robustez.

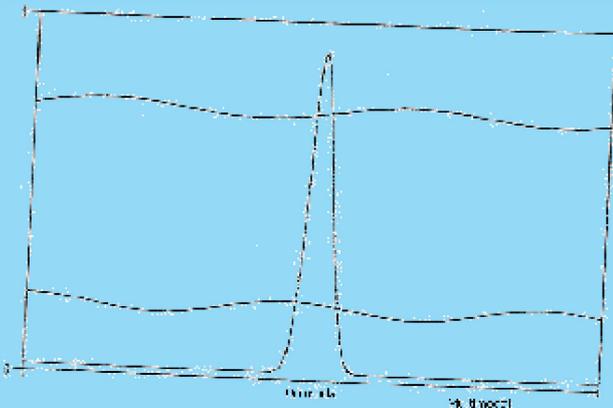


Figura 1.4 Algunos esquemas tradicionales trabajan bien en problemas de dominio limitado.



Las metas de la optimización

Antes de comenzar a estudiar los mecanismos y la potencia de un algoritmo genético simple, se debe hacer claridad sobre las metas cuando se dice, que se quiere optimizar una función o un proceso. Un punto de vista convencional es bien presentado por Beightler, Phillips y Wilde (1979): El anhelo del hombre por la perfección encuentra expresión en la teoría de la optimización. Esta estudia cómo describir y obtener lo que sea mejor. Una vez que se sabe cómo medir y alterar lo que es bueno y lo que es malo. La teoría de la optimización abarca el estudio cuantitativo de lo óptimo y los métodos para encontrarlos.

Así, la optimización busca mejorar la eficiencia hacia algún punto o puntos óptimos. Note que esta definición tiene dos partes: 1) Se busca mejorar un acercamiento a algo, y 2) buscamos el punto óptimo. Hay una clara distinción entre el proceso de mejorar y la meta o el mismo óptimo.

Considere la toma de decisiones, por ejemplo, de un hombre de negocios. ¿Cómo hacer para juzgar su decisión?, ¿Qué criterio utilizar para decidir si él ha hecho un buen o mal trabajo? Usualmente se dice que él ha hecho bien las cosas cuando toma selecciones adecuadas en el tiempo y con los recursos adjudicados. La

Fuente:
http://2.bp.blogspot.com/_W2Je2sQK31/RohODgCdGI/AAAAAAAAAc/tSx3MPegqCs/S76o/Algoritmos_Gen_ticos_Herr1.jpg

bondad es una virtud relativa a su competencia. ¿Él produjo la mejor decisión? ¿Lo que él consiguió fue para ser más eficiente? ¿Con mejor promoción? Nunca se juzga a un hombre de negocios con el criterio de que haya alcanzado lo mejor. La perfección de un jefe es hacer todo de la forma más rigurosa. Como un juicio se concluye que la convergencia hacia lo mejor no es un problema del negocio, sino que se está solamente interesado en hacer lo mejor relativamente. Así, si se desean más herramientas de optimización como humanos, se debe llevar a un reordenamiento de las

prioridades de optimización. La meta más importante de la optimización es la perfección o mejoramiento. Siempre habrán preguntas como, se puede obtener algo mejor?, los niveles satisfactorios (Simón, 1969) de eficiencia se pueden lograr rápidamente?

El logro de el óptimo es bastante menos importante para los sistemas complejos? Las cosas deben ser buenas para que sean perfectas? Entre tanto solo se puede esforzarse para mejorar cada vez más.



Qué son los Algoritmos Genéticos Simples (AGS)

En su trabajo, Holland propone una manera de seleccionar individuos y de cruzarlos. Actualmente existen muchas otras propuestas, pero las de Holland constituyen aún hoy la base de muchos desarrollos teóricos y prácticos este tema. Goldberg retomó el algoritmo de Holland y lo popularizó llamándolo

algoritmo genético simple (AGS). En éste se considera que los códigos genéticos están en binario. Explicado sin detalle, el proceso de un AGS es:

1. Decidir cómo codificar el dominio del problema.
2. Generar un conjunto aleatorio (población inicial) de N posibles soluciones codificadas de acuerdo al problema. A ésta se le llamará la población actual.
3. Calificar cada posible solución (individuo) de la población actual.
4. Seleccionar dos individuos de la población actual con una probabilidad proporcional a su calificación.
5. Lanzar una moneda al aire con probabilidad de cruce p_c si cae cara.
6. Si cayó cara mezclar los códigos de los dos individuos seleccionados para formar dos híbridos, a los que llamaremos nuevos individuos.
7. Si cayó cruz llamamos a los individuos seleccionados nuevos individuos.
8. Por cada bit de cada nuevo individuo lanzar otra moneda al aire con probabilidad de mutación p_m si cae cara.
9. Si cae cara cambiar el bit en turno por su complemento.
10. Si cae cruz el bit permanece inalterado.

11. Incluir a los dos nuevos individuos en una nueva población.
12. Si la nueva población tiene ya N individuos, llamarla población actual y regresar al paso 3, a menos que se cumpla alguna condición de terminación.
13. Si no, regresar al paso 4.

En el algoritmo se utiliza el término "lanzar una moneda al aire" para decir un experimento de Bernoulli (aquel en el que pueden ocurrir exclusivamente dos eventos posibles, uno con probabilidad p y otro con probabilidad $1-p$).

La condición de terminación, a la que se hace referencia en el paso 12, puede definirse de muchas maneras. Se puede fijar un número máximo de generaciones que debe ejecutar el algoritmo, o puede decidirse hacer alto cuando la mayoría de la población, digamos el 85%, tenga una calificación que esté dentro de 0.6 desviaciones estándar de la media. En fin, opciones hay muchas, generalmente depende del problema o de las preferencias personales la decisión de cuándo es conveniente detenerse.

En el paso 4, se menciona que hay que seleccionar dos individuos con probabilidad proporcional a su calificación. Este tipo de selección proporcional es también llamado de "ruleta" (roulette wheel selection) por lo siguiente: supóngase que se suman las calificaciones de todos los individuos de la población y esta suma es considerada el 100% de una circunferencia. Luego, a cada individuo se le asigna una longitud de la misma según su aportación a la suma de las calificaciones. Aunque existen otras variantes, básicamente la idea presentada en este escrito es la base fundamental para la comprensión de lo que son los algoritmos genéticos, como se muestra a continuación y donde se observa su desempeño en el ámbito discreto.

Fuente:
http://1.bp.blogspot.com/_KKPGWfonFms/TDjq5fulGCI/AAAAAAAAAU/8KlmrENkhr8/s320/algorithm.jpg

• Algoritmo genético simple

La forma más simple de un AG involucra tres tipos de operadores: selección, cruce y mutación. A continuación se da una explicación de cada uno.

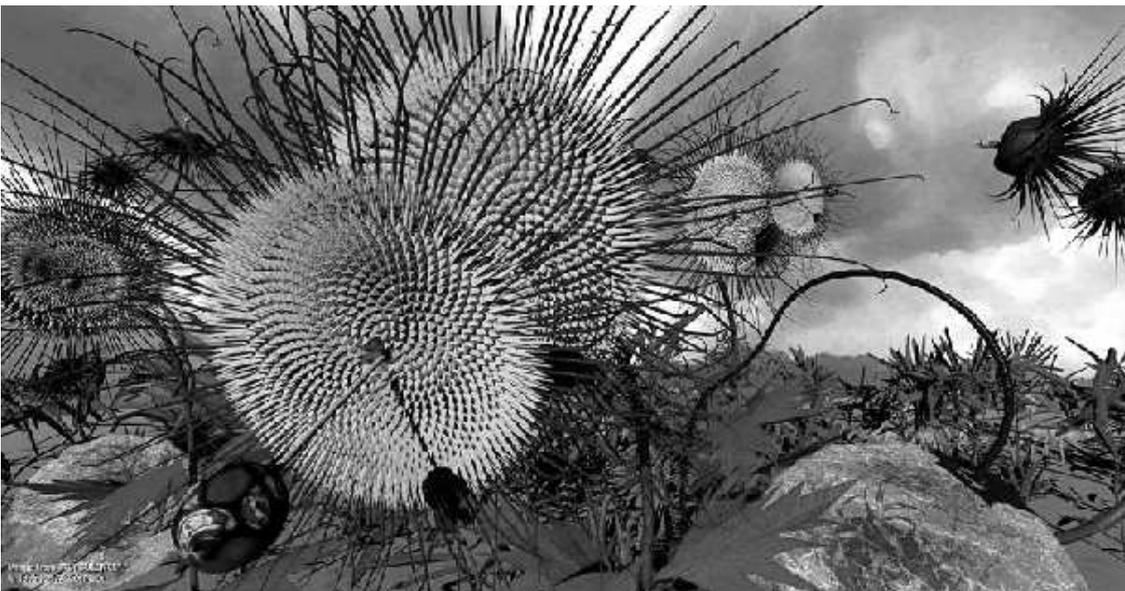
Selección: Este operador selecciona cromosomas de la población para posteriormente permitir que se reproduzcan. Esta selección está basada en la aptitud del cromosoma: entre más apto más probabilidad tiene de ser seleccionado, es decir, mayor probabilidad de poder contribuir con uno o más descendientes en la siguiente generación.

Cruce: Este operador escoge aleatoriamente una posición e intercambia la subsecuencia antes y después de la posición, entre dos cromosomas para crear dos hijos. Por ejemplo, las cadenas 10000100 y 11111111 pueden ser cruzadas después de la tercera posición en cada una para producir los hijos 10011111 y 11100100. El operador de cruce realiza la recombinación entre dos organismos de un solo cromosoma.

Mutación: Este operador cambia aleatoriamente algunos de los bits en un cromosoma; así, permite hacer una exploración completa para evitar concentración de los genes más aptos en máximos locales. Por ejemplo, la cadena 00000100 puede ser mutada en su segunda posición y producir 01000100. La mutación puede ocurrir en cada una de las posiciones de la cadena con alguna probabilidad, usualmente esta probabilidad es muy pequeña, como 0.001.

Dada una definición clara del problema a resolver y una representación en cadena de símbolos para las soluciones candidatas, este ejemplo trabaja de la siguiente manera:

1. Se comienza con una población generada aleatoriamente de k cromosomas de 1 bits (soluciones candidatas para la solución del problema).
2. Se calcula la aptitud $f(x)$ de cada cromosoma x en la población.
3. Se repiten los siguientes pasos hasta que k hijos hayan sido creados (generalmente los valores de k y 1 están en el rango de 50 a 1000):
 - Selecciona un par de cromosomas de la población actual (estas cromosomas serán los padres). La probabilidad de selección estará dada de acuerdo a la aptitud de cada cromosoma. La selección se hace sin reemplazo.



Fuente fotográfica:
<http://carlosreynoso.com.ar/wp-content/plugins/download-monitor/download.php>



Fuente fotográfica:
<http://cerebricos.blogia.com/upload/20100214201021-algoritmos-geneticos.jpg>

- Con probabilidad p_c (probabilidad de cruce o razón de cruce), se cruzan el par de cromosomas en un punto escogido aleatoriamente (con una probabilidad uniforme) para formar dos hijos. Si no se efectúa el cruce los hijos son copias exactas de sus respectivos padres.
 - De acuerdo con una probabilidad p_m , (probabilidad de mutación o razón de mutación), se elige sustituir el símbolo de cada posición de las cadenas de los hijos, por otro cualquiera. Así se obtienen los cromosomas de la nueva población.
4. Se reemplaza la población actual por la población nueva.
 5. Se regresa al paso 2.

Cada iteración de este proceso se llama generación M . Un AG se itera generalmente hasta valores de $M = \{50,500\}$ o inclusive más generaciones dependiendo de la complejidad del problema, de la codificación del cromosoma y la evaluación de la función de aptitud. Un conjunto completo de generaciones es llamado ejecución; al final de la ejecución se encuentran usualmente uno o más cromosomas con una muy buena aptitud en la población.

Como un ejemplo más detallado de un AG simple, suponga que l (longitud de la cadena) es 8, que $f(x)$ es igual al número de unos en la cadena de bits x , que k (tamaño de la población A, B, C y D) es 4, que $P_c = 0.7$ y $P_m = 0.001$. La

población inicial (generada aleatoriamente) se muestra en la tabla 1.

El método de selección es la selección proporcional a la aptitud, esto es equivalente a lo que los biólogos llaman “viabilidad de selección”. Cabe notar que incluso los individuos con muy bajo porcentaje de aptitud también tienen la posibilidad de ser seleccionados.

Etiqueta	Cromosoma	Aptitud
A	00000110	2
B	11101110	6
C	00100000	1
D	00110100	3

Tabla 1. Población inicial generada aleatoriamente, para un AG simple.

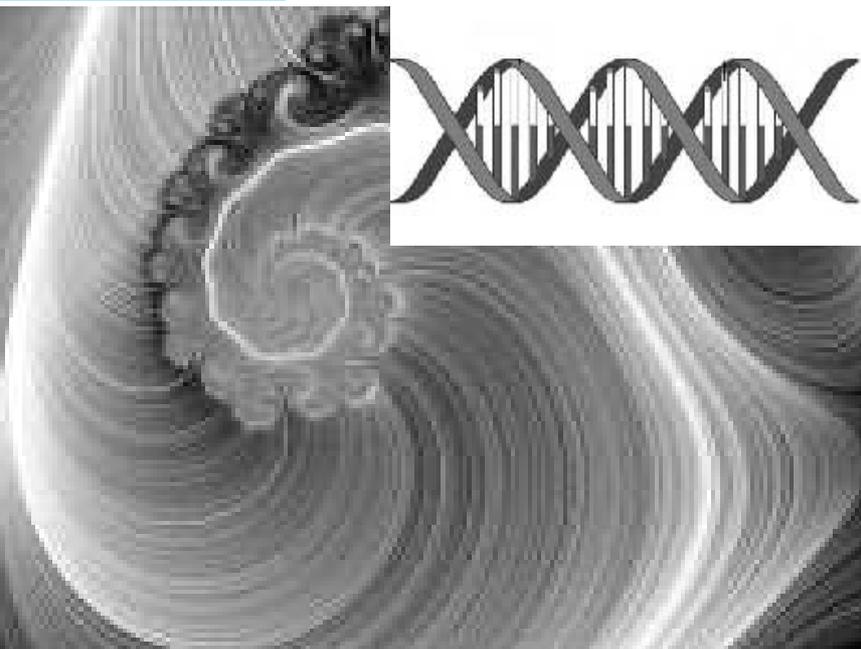
Una vez se seleccionan los padres, se combina su contenido genético (es decir, sus cromosomas) con una probabilidad P_c para formar dos hijos. Suponga que se efectúa un cruce entre B y D después de la primera posición así que se forman $E=10110100$ y $F=01101110$, y que los padres B y C no se cruzan, luego, cada hijo es sujeto a mutación en cada una de sus posiciones con probabilidad p_m ; por ejemplo, suponga que el hijo E se muta en la sexta posición para formar $E'=10110000$ los hijos F y C no se mutan y el hijo B se muta en la primera posición para formar $B'=01101110$. La nueva población se muestra en la tabla 2.

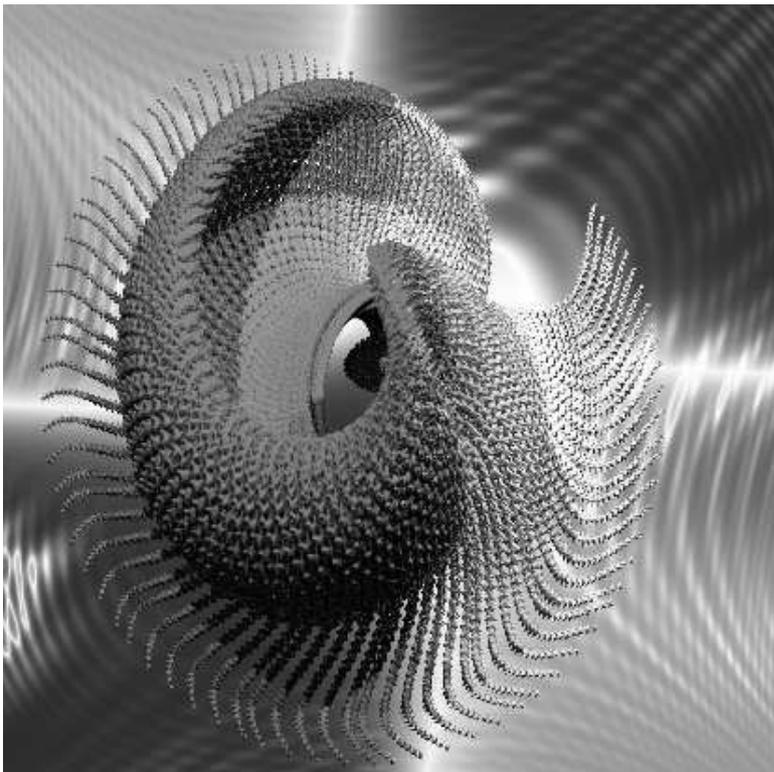
Etiqueta	Cromosoma	Aptitud
E'	10110000	3
F	01101110	5
C	00100000	1
B'	01101110	5

Tabla 2. Población descendiente de la primera generación.

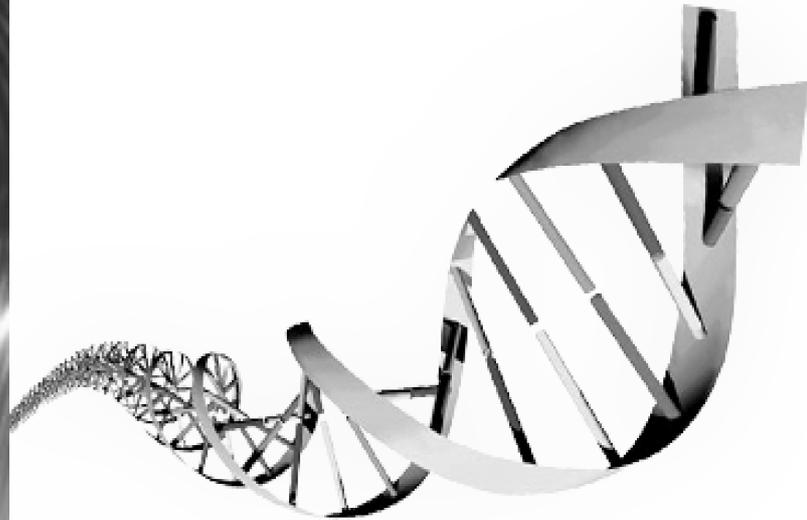
Nótese que en la nueva población, aunque la mejor cadena (la cual tenía una aptitud de 6) se ha perdido, el promedio de aptitud de la población, aumentó de $12/4$ a $14/4$. Iterando este procedimiento se encontrará eventualmente una cadena compuesta de solo unos.

Fuente:
http://2.bp.blogspot.com/_4WY6HZkSj3k/SPIr5elk4RI/AAAAAAAAAGA/EghlbtqDqxc/s200/fractal-shell_glow.jpg





Fuente: http://3.bp.blogspot.com/_oWwzN57oyg8/TIEs3twFRtl/AAAAAAAAADH8/dPtBNimtdcl/s1600/runnerfrog_-_3D_abstracta_%23136.jpg



Fuente: <http://just4cool.files.wordpress.com/2008/12/dna.png>

Referencias

AMADOR, N. J., PINEDA, W., Diseño y Desarrollo de Algoritmos y Sistemas de Control por Clonación Artificial de un sensor de viscosidad. Tesis de Maestría en Ciencias Computacionales, 193 p., 2006.

GOLDBERG, D.E. Genetic Algorithms in Search, Optimization & Machine Learning. Reading: Addison-Wesley, 1989.

MITCHELL, M. An Introduction To Genetic Algorithms. Eight edition. Cambridge: MIT Press, 2002.

MUÑOZ, A.F., Aplicación de los algoritmos genéticos en la identificación y control de bioprocesos por clonación artificial. IEEE Transactions on Systems, Man, and Cybernetic V. 19 No.2 58-76, 1998.

Jairo Amador Niño. Licenciado en Matemáticas UPTC. Ingeniero de Sistemas UAN. Especialista en Telemática UNIBOYACÁ. Magíster en Ciencias Computacionales UNAB-TEC México. Docente Universidad Santo Tomás.



Fuente: http://4.bp.blogspot.com/_yaHiNrHfiNg/TNyrzu4GqxI/AAAAAAAAAa0/qH6Ou3-pWfw/s1600/forense.jpg