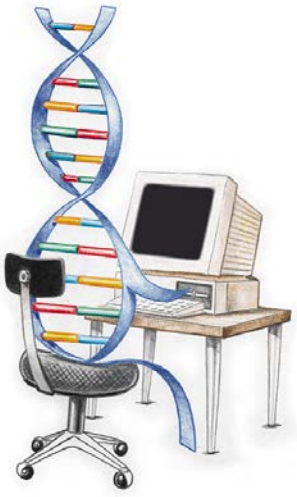
Cátedra de: ALGORITMOS GENÉTICOS

"INTRODUCCIÓN A LOS ALGORITMOS GENÉTICOS"



Profesora: Ing. Daniela Díaz

Universidad Tecnológica Nacional
Facultad Regional Rosario

Año: 2013

Introducción

John Holland desde pequeño, se preguntaba cómo logra la naturaleza, crear seres cada vez más perfectos (aunque, como se ha visto, esto no es totalmente cierto, o en todo caso depende de qué entienda uno por *perfecto*). Lo curioso era que todo se lleva a cabo a base de interacciones locales entre individuos, y entre estos y lo que les rodea. No sabía la respuesta, pero tenía una cierta idea de como hallarla: tratando de hacer pequeños modelos de la naturaleza, que tuvieran alguna de sus características, y ver cómo funcionaban, para luego extrapolar sus conclusiones a la totalidad. De hecho, ya de pequeño hacía simulaciones de batallas célebres con todos sus elementos: copiaba mapas y los cubría luego de pequeños ejércitos que se enfrentaban entre sí.

En realidad el objetivo de Holland era lograr que las computadoras aprendieran por sí mismas. La técnica que inventó se le llamó originalmente "planes reproductivos", pero se hizo popular bajo el nombre "algoritmo genético" tras la publicación de su libro en 1975

En los años 50 entró en contacto con los primeros ordenadores, donde pudo llevar a cabo algunas de sus ideas, aunque no se encontró con un ambiente intelectual fértil para propagarlas. Fue a principios de los 60, en la Universidad de Michigan en Ann Arbor, donde, dentro del grupo *Logic of Computers,* sus ideas comenzaron a desarrollarse y a dar frutos. Y fue, además, leyendo un libro escrito por un biólogo evolucionista, R. A. Fisher, titulado *La teoría genética de la selección natural,* como comenzó a descubrir los medios de llevar a cabo sus propósitos de comprensión de la naturaleza. De ese libro aprendió que la evolución era una forma de adaptación más potente que el simple aprendizaje, y tomó la decisión de aplicar estas ideas para desarrollar programas bien adaptados para un fin determinado.

En esa universidad, Holland impartía un curso titulado *Teoría de sistemas adaptativos.* Dentro de este curso, y con una participación activa por parte de sus estudiantes, fue donde se crearon las ideas que más tarde se convertirían en los algoritmos genéticos.

Por tanto, cuando Holland se enfrentó a los algoritmos genéticos, los objetivos de su investigación fueron dos:

- imitar los procesos adaptativos de los sistemas naturales, y
- diseñar sistemas artificiales (normalmente programas) que retengan los mecanismos importantes de los sistemas naturales.

Unos 15 años más adelante, **David Goldberg,** actual delfín de los algoritmos genéticos, conoció a Holland, y se convirtió en su estudiante. Golberg era un ingeniero industrial trabajando en diseño de *pipelines*, y fue uno de los primeros que trató de aplicar los algoritmos genéticos a problemas industriales. Aunque Holland trató de disuadirle, porque pensaba que el problema era excesivamente complicado como para aplicarle algoritmos genéticos, Goldberg consiguió lo que quería, escribiendo un algoritmo genético en un ordenador personal Apple II. Estas y otras aplicaciones creadas por estudiantes de Holland convirtieron a los algoritmos genéticos en un campo con base suficiente aceptado para celebrar la primera conferencia en 1985, ICGA´85. Tal conferencia se sigue celebrando bianualmente.

Antecedentes
El algoritmo genético es una técnica de búsqueda basada en la teoría de la evolución de Darwin, que ha cobrado tremenda popularidad en todo el mundo durante los últimos años.

En los últimos años, la comunidad científica internacional ha mostrado un creciente interés en una nueva técnica de búsqueda basada en la teoría de la evolución y que se conoce como el ***algoritmo genético***. 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 de un individuo (unidad básica de

codificación de cada uno de los atributos de un ser vivo), y que sus atributos más deseables (los que le permiten adaptarse mejor a su entorno) se transmiten a sus descendientes cuando éste se reproduce sexualmente.

Anatomía de un algoritmo genético simple

Los *algoritmos genéticos* son métodos sistemáticos para la resolución de problemas de búsqueda y optimización que aplican a estos los mismos métodos de la evolución biológica: selección basada en la población, reproducción sexual y mutación.

Los Algoritmos Genéticos (AGs) son métodos adaptativos que pueden usarse para resolver problemas de búsqueda y optimización. Están basados en el proceso genético de los organismos vivos. A lo largo de las generaciones, las poblaciones evolucionan en la naturaleza de acorde con los principios de la selección natural y la supervivencia de los más fuertes, postulados por Darwin. Por imitación de este proceso, los Algoritmos Genéticos son capaces de ir creando soluciones para problemas del mundo real. La evolución de dichas soluciones hacia valores óptimos del problema depende en buena medida de una adecuada codificación de las mismas.

Un algoritmo genético consiste en una función matemática o una rutina de software que toma como entradas a los ejemplares y retorna como salidas cuales de ellos deben generar descendencia para la nueva generación.

Versiones más complejas de algoritmos genéticos generan un ciclo iterativo que directamente toma a la especie (el total de los ejemplares) y crea una nueva generación que reemplaza a la antigua una cantidad de veces determinada por su propio diseño. Una de sus características principales es la de ir perfeccionando su propia heurística en el proceso de ejecución, por lo que no requiere largos períodos de entrenamiento especializado por parte del ser humano, principal defecto de otros métodos para solucionar problemas, como los Sistemas Expertos.

Algunas definiciones nos dicen que:

“Los ALGORITMOS GENÉTICOS son una rama de la Inteligencia Artificial que imita la evolución biológica para resolver problemas de búsqueda y optimización. Se basan en el proceso genéticos de los organismos vivos.”

“Los ALGORITMOS GENÉTICOS son algoritmos que trabajan con una población de individuos, cada uno de los cuales representa una solución factible a determinados problemas. Son algoritmos de búsqueda múltiple, es decir algoritmos que dan varias soluciones, unas mejores que otras.”

Una definición bastante completa de un algoritmo genético es la propuesta por John Koza:

"Es 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 suele ser una cadena de caracteres (letras o números) de longitud fija que se ajusta al modelo de las cadenas de cromosomas, y se les asocia con una cierta función matemática que refleja su aptitud. "

Los *algoritmos genéticos* son métodos de optimización, que tratan de resolver problemas como por ejemplo hallar $(x_i,...,x_n)$ tales que $F(x_i,...,x_n)$ sea máximo.

En un *algoritmo genético*, tras parametrizar el problema en una serie de variables, $(x_i,...,x_n)$ se codifican en un cromosoma. Todos los operadores utilizados por un *algoritmo genético* se aplicarán sobre estos cromosomas, o sobre poblaciones de ellos. En el *algoritmo genético* va implícito el método para resolver el problema; son solo parámetros de tal método los que están codificados, a diferencia de otros algoritmos evolutivos como la programación genética. Hay que tener en cuenta que un algoritmo genético es independiente del problema, lo cual lo hace un algoritmo *robusto*, por ser útil para cualquier problema, pero a la vez *débil*, pues no está especializado en ninguno.

Las soluciones codificadas en un cromosoma *compiten* para ver cuál constituye la mejor solución (aunque no necesariamente la mejor de todas las soluciones posibles). El *ambiente,* constituido por las otras camaradas soluciones, ejercerá una presión selectiva sobre la población, de forma que sólo los mejor adaptados (aquellos que resuelvan mejor el problema) sobrevivan o leguen su material genético a las siguientes generaciones, igual que en la evolución de las especies. La diversidad genética se introduce mediante mutaciones y reproducción sexual.

En la Naturaleza lo único que hay que optimizar es la supervivencia, y eso significa a su vez maximizar diversos factores y minimizar otros. Un *algoritmo genético*, sin embargo, se usará habitualmente para optimizar sólo una función, no diversas funciones relacionadas entre sí simultáneamente. La optimización que busca diferentes objetivos simultáneamente, denominada multimodal o multiobjetivo, también se suele abordar con un algoritmo genético especializado.

Por lo tanto, un algoritmo genético consiste en hallar de qué parámetros depende el problema, codificarlos en un cromosoma, y se aplican los métodos de la evolución: selección y reproducción sexual con intercambio de información y alteraciones que generan diversidad.

Al concepto de Algoritmos lo podemos definir como una serie de pasos organizados que describe el proceso que se debe seguir, para dar solución a un problema específico.
En el año 1970, como se explicó anteriormente de la mano de John Henry Holland, surgió una de las líneas más prometedoras de la inteligencia artificial, la de los *algoritmos genéticos* los cuales se inspiran en la evolución biológica y su base genético-molecular.

Estos algoritmos hacen evolucionar una población de individuos sometiéndola a acciones aleatorias semejantes a las que actúan en la evolución biológica (mutaciones y recombinaciones genéticas), así como también a una selección de acuerdo con algún criterio, en función del cual se decide cuáles son los individuos más adaptados, que sobreviven, y cuáles los menos aptos, que son descartados. También son denominados algoritmos evolutivos, e incluyen las estrategias de evolución, la programación evolutiva y la programación genética.

Dentro de la programación genética se han logrado avances curiosos entre los cuales cabe mencionar el ocurrido en 1999, cuando por primera vez en la historia, se concedió una patente a un invento no realizado directamente por un ser humano: se trata de una *antena* de forma extraña, pero que funciona perfectamente en las condiciones a las que estaba destinada. No hay, sin embargo, nada injusto en el hecho de que el autor del algoritmo genético del que salió la forma de la antena se haya atribuido la autoría de la patente, pues él escribió el programa e ideó el criterio de selección que condujo al diseño patentado.

Podemos decir que un algoritmo genético es un método de búsqueda dirigida basada en probabilidad. Bajo una condición muy débil (que el algoritmo mantenga elitismo, es decir, guarde siempre al mejor elemento de la población sin hacerle ningún cambio) se puede demostrar que el algoritmo converge en probabilidad al óptimo. En otras palabras, al aumentar el número de iteraciones, la probabilidad de tener el óptimo en la población tiende a 1 (uno).

Codificación de las variables

Los *algoritmos genéticos* requieren que el conjunto se codifique en un ***cromosoma***. Cada cromosoma tiene varios genes, que corresponden a sendos parámetros del problema. Para poder trabajar con estos genes en el ordenador, es necesario codificarlos en una *cadena*, es decir, una ristra de símbolos (números o letras) que generalmente va a estar compuesta de 0s y 1s.

Por ejemplo, en esta cadena de bits, el valor del parámetro *p1* ocupará las posiciones 0 a 2, el *p2* las 3 a 5, etcétera. El número de bits usado para cada parámetro dependerá de la precisión que se quiera en el mismo o del número de opciones posibles que tenga ese parámetro.

Hay otras codificaciones posibles, usando alfabetos de diferente cardinalidad; sin embargo, uno de los resultados fundamentales en la teoría de algoritmos genéticos, el *teorema de los esquemas*, afirma que la codificación óptima, es decir, aquella sobre la que los algoritmos genéticos funcionan mejor, es aquella que tiene un alfabeto de cardinalidad[1] 2.

Se está codificando cada parámetro como un número entero de *n* bits. En realidad, se puede utilizar cualquier otra representación interna: bcd, código *Gray* y codificación en forma de números reales, por ejemplo.

La mayoría de las veces, una codificación correcta es la clave de una buena resolución del problema. Generalmente, la regla heurística que se utiliza es la llamada *regla de los bloques de construcción*, es decir, parámetros relacionados entre sí deben de estar cercanos en el cromosoma. Por ejemplo, si queremos codificar los pesos de una red neuronal, una buena elección será poner juntos todos los pesos que salgan de la misma neurona de la capa oculta.

Se puede llevar a usar cromosomas bidimensionales, o tridimensionales, o con relaciones entre genes que no sean puramente lineales de vecindad. En algunos casos, cuando no se conoce de antemano el número de variables del problema, caben dos opciones: codificar también el número de variables, fijando un número máximo, o bien, lo cual es mucho más natural, crear un cromosoma que pueda variar de longitud. Para ello, claro está, se necesitan operadores genéticos que alteren la longitud.

Normalmente, la codificación es estática, pero en casos de optimización numérica, el número de bits dedicados a codificar un parámetro puede variar, o incluso lo que representen los bits dedicados a codificar cada parámetro. Algunos paquetes de algoritmos genéticos adaptan automáticamente la codificación según van convergiendo los bits menos significativos de una solución.

[1] La cardinalidad de un conjunto se representa con el símbolo # y corresponde al número de elementos que tiene el conjunto.

Algoritmo genético: su funcionamiento

Los algoritmos genéticos involucran una **función objetivo** de naturaleza diversa sobre la cual debe encontrarse un mínimo o un máximo según el problema planteado. Al conjunto de posibles soluciones lo llamaremos **espacio solución**. Cada elemento del espacio solución, llamado **cromosoma**, es una tira de **genes**.

El algoritmo necesita, para comenzar a funcionar, una población llamada **población inicial** formada por una cantidad N de cromosomas elegidos al azar.
Estos cromosomas tienen en correspondencia los valores de la función objetivo, y además otros valores suministrados por otra función llamada **fitness** que mide su potencial como padres.
La función **fitness** puede coincidir con la función objetivo en algunos casos, mientras que en otros se define de manera distinta dependiendo del problema en cuestión.

En general los espacios solución que utilizamos son muy amplios por lo que la posibilidad de encontrar la solución en la población inicial elegida al azar, es casi imposible. El proceso continúa, entonces, tratando de obtener otra población mejorada (hijos), con la misma cantidad de elementos N que la inicial.

Para comenzar la competición, se generan aleatoriamente una serie de cromosomas. El algoritmo genético procede de la forma siguiente:

1. Evaluar la puntuación (*fitness*) de cada uno de los genes.
2. Permitir a cada uno de los individuos reproducirse, de acuerdo con su puntuación.
3. Emparejar los individuos de la nueva población, haciendo que intercambien material genético, y que alguno de los bits de un gen se vea alterado debido a una *mutación* espontánea.

Cada uno de los pasos consiste en una actuación sobre las cadenas de bits, es decir, la aplicación de un ***operador*** a una cadena binaria. Se les denominan, por razones obvias, ***operadores genéticos***, y hay tres principales: ***selección***, ***crossover*** o recombinación y ***mutación***; aparte de otros operadores genéticos no tan comunes.

Selección: un operador genético extrae, aleatoriamente, pares de elementos de la población inicial.
Crossover: cada par de elementos seleccionados, denominados padres, se somete al control mediante una rutina llamada **probabilidad de crossover** que indica si se realizará o no el crossover. Si la respuesta es afirmativa, hay operadores genéticos que lo realizan logrando finalmente la obtención de dos hijos. Si la respuesta es negativa, los padres pasan directamente a ser hijos.
Mutación: cada uno de estos hijos, a su vez, se controla con otra rutina llamada **probabilidad de mutación**. En caso de ser afirmativa, este hijo sufre una mutación con un operador adecuado.

Un algoritmo genético tiene también una serie de parámetros que se tienen que fijar para cada ejecución, como los siguientes:

- **Tamaño de la población**: debe de ser suficiente para garantizar la diversidad de las soluciones, y, además, tiene que crecer más o menos con el número de bits del cromosoma, aunque nadie ha aclarado cómo tiene que hacerlo. Por supuesto, depende también del ordenador en el que se esté ejecutando.
- **Condición de terminación:** lo más habitual es que la condición de terminación sea la convergencia del algoritmo genético o un número prefijado de generaciones.

Evaluación y selección

Durante la evaluación, se decodifica el gen, convirtiéndose en una serie de parámetros de un problema, se halla la solución del problema a partir de esos parámetros, y se le da una puntuación a esa solución en función de lo cerca que esté de la mejor solución. A esta puntuación se le llama *fitness*. El fitness determina siempre los cromosomas que se van a reproducir, y aquellos que se van a eliminar, pero hay varias formas de considerarlo para seleccionar la población de la siguiente generación:

- Usar el orden, o rango, y hacer depender la probabilidad de permanencia o evaluación de la posición en el orden.
- En algunos casos, el fitness no es una sola cantidad, sino diversos números, que tienen diferente consideración. Basta con que tal fitness forme un orden parcial, es decir, que se puedan comparar dos individuos y decir cuál de ellos es mejor. Esto suele suceder cuando se necesitan optimizar varios objetivos.

Una vez evaluado el fitness, se tiene que crear la nueva población teniendo en cuenta que los *buenos* rasgos de los mejores se transmitan a esta. Para ello, hay que seleccionar a una serie de individuos encargados de tan ardua tarea. Y esta selección, y la consiguiente reproducción, se puede hacer de las siguientes formas principales:

- ***Basado en el rango*:** en este esquema se mantiene un porcentaje de la población, generalmente la mayoría, para la siguiente generación. Se coloca toda la población por orden de fitness, y los *M* menos dignos son eliminados y sustituidos por la descendencia de alguno de los *M* mejores con algún otro individuo de la población. A este esquema se le pueden aplicar otros criterios; por ejemplo, se crea la descendencia, y esta sustituye al más parecido entre los perdedores. Esto se denomina *crowding*, y fue introducido por **DeJong.**. Una variante de este es el muestreado estocástico universal, que trata de evitar que los individuos con más fitness copen la población; en vez de dar la vuelta a una ruleta con una ranura, da la vuelta a la ruleta con N ranuras, tantas como la población; de esta forma, la distribución estadística de descendientes en la nueva población es más parecida a la real.
- ***Rueda de ruleta*:** se crea un *pool* genético formado por cromosomas de la generación actual, en una cantidad proporcional a su fitness. Si la proporción hace que un individuo domine la población, se le aplica alguna operación de escalado. Dentro de este *pool*, se escogen parejas aleatorias de cromosomas y se emparejan, sin importar incluso que sean del mismo progenitor (para eso están otros operadores, como la mutación). Hay otras variantes: por ejemplo, en la nueva generación se puede incluir el mejor representante de la generación actual. En este caso, se denomina método *elitista*.
- ***Selección de torneo*:** se escogen aleatoriamente un número *T* de individuos de la población, y el que tiene puntuación mayor se reproduce, sustituyendo en su descendencia al que tiene menor puntuación.

Crossover

Consiste en el intercambio de material genético entre dos cromosomas, a veces más de dos. El ***crossover*** es el principal operador genético, hasta el punto que se puede decir que no es un algoritmo genético si no tiene ***crossover***, y, sin embargo, puede serlo perfectamente sin mutación, según descubrió Holland.

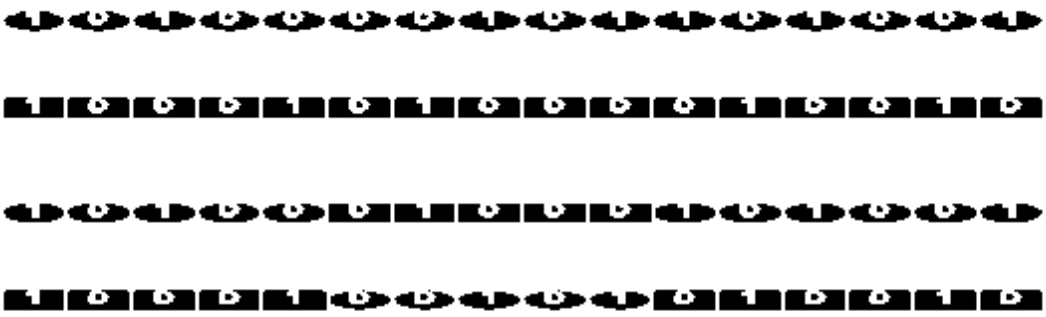
Para aplicar el crossover, entrecruzamiento o recombinación, se escogen aleatoriamente dos miembros de la población. No pasa nada si se emparejan dos descendiente de los mismos padres; ello garantiza la perpetuación de un individuo con buena puntuación (y, además, algo parecido ocurre en la realidad; es una práctica utilizada, por ejemplo, en la cría de ganado, llamada *inbreeding*, y destinada a potenciar ciertas características frente a otras). Sin embargo,

si esto sucede demasiado a menudo, puede crear problemas: toda la población puede aparecer dominada por los descendientes de algún gen, que, además, puede tener caracteres no deseados. Esto se suele denominar en otros métodos de optimización *atranque en un mínimo local*, y es uno de los principales problemas con los que se enfrentan los que aplican algoritmos genéticos.

El teorema de los esquemas, se basa en la noción de *bloques de construcción*. Una buena solución a un problema está constituida por unos buenos bloques, igual que una buena máquina está hecha por buenas piezas. El crossover es el encargado de mezclar bloques buenos que se encuentren en los diversos progenitores, y que serán los que den a los mismos una buena puntuación. La presión selectiva se encarga de que sólo los buenos bloques se perpetúen, y poco a poco vayan formando una buena solución. El *teorema de los esquemas* viene a decir que la cantidad de *buenos bloques* se va incrementando con el tiempo de ejecución de un algoritmo genético, y es el resultado teórico más importante en algoritmos genéticos.

El intercambio genético se puede llevar a cabo de muchas formas, pero hay dos grupos principales

- ***Crossover n-puntos:*** los dos cromosomas se cortan por n puntos, y el material genético situado entre ellos se intercambia. Lo más habitual es un crossover de un punto o de dos puntos.



- ***Crossover uniforme*:** se genera un patrón aleatorio de 1s y 0s, y se intercambian los bits de los dos cromosomas que coincidan donde hay un 1 en el patrón. O bien se genera un número aleatorio para cada bit, y si supera una determinada probabilidad se intercambia ese bit entre los dos cromosomas.

- ***Crossover especializados:*** en algunos problemas, aplicar aleatoriamente el crossover da lugar a cromosomas que codifican soluciones inválidas; en este caso hay que aplicar el crossover de forma que genere siempre soluciones válidas. Un ejemplo de estos son los operadores de crossover usados en el problema del viajante.

Mutación

En la Evolución, una ***mutación*** es un suceso bastante poco común (sucede aproximadamente una de cada mil replicaciones). En la mayoría de los casos las ***mutaciones*** son letales, pero en promedio, contribuyen a la diversidad genética de la especie. En un algoritmo genético tendrán el mismo papel, y la misma frecuencia (es decir, muy baja).

Una vez establecida la frecuencia de mutación, por ejemplo, uno por mil, se examina cada bit de cada cadena cuando se vaya a crear la nueva criatura a partir de sus padres (normalmente se hace de forma simultánea al crossover). **Si un número generado aleatoriamente está por debajo de esa probabilidad,** se **cambiará el bit** (es decir, de 0 a 1 o de 1 a 0). Si no, se dejará como está. Dependiendo del número de individuos que haya y del número de bits por individuo, puede resultar que las mutaciones sean extremadamente raras en una sola generación.

No hace falta decir que no conviene abusar de la ***mutación***. Es cierto que es un mecanismo generador de diversidad, y, por tanto, la solución cuando un algoritmo genético está estancado, pero también es cierto que reduce el algoritmo genético a una búsqueda aleatoria. Siempre es más conveniente usar otros mecanismos de generación de diversidad, como aumentar el tamaño de la población, o garantizar la aleatoriedad de la población inicial.

Este operador, junto con la anterior y el método de selección de ruleta, constituyen un *algoritmo genético simple.*

Otros operadores

No se usan en todos los problemas, sino sólo en algunos, y en principio su variedad es infinita. Generalmente son operadores que exploran el espacio de soluciones de una forma más ordenada, y que actúan más en las últimas fases de la búsqueda, en la cual se pasa de soluciones "casi buenas" a "buenas" soluciones.

Cromosomas de longitud variable

Hasta ahora se han descrito cromosomas de longitud fija, donde se conoce de antemano el número de parámetros de un problema. Pero hay problemas en los que esto no sucede. Por ejemplo, en un problema de clasificación, donde dado un vector de entrada, queremos agruparlo en una serie de clases, podemos no saber siquiera cuantas clases hay. O en diseño de redes neuronales, puede que no se sepa (de hecho, nunca se sabe) cuántas neuronas se van a necesitar. Por ejemplo, en un perceptrón hay reglas que dicen cuantas neuronas se deben de utilizar en la capa oculta; pero en un problema determinado puede que no haya ninguna regla heurística aplicable; tendremos que utilizar los algoritmos genéticos para hallar el número óptimo de neuronas. En este caso, si utilizamos una *codificación fregona*, necesitaremos un locus[2] para cada neurona de la capa oculta, y el número de locus variará dependiendo del número de neuronas de la capa oculta.

En estos casos, necesitamos dos operadores más: *añadir* y *eliminar*. Estos operadores se utilizan para añadir un gen, o eliminar un gen del cromosoma. La forma más habitual de añadir un locus es *duplicar* uno ya existente, el cual sufre mutación y se añade al lado del anterior. En este caso, los operadores del algoritmo genético simple (selección, mutación, crossover) funcionarán de la forma habitual, salvo, claro está, que sólo se hará crossover en la zona del cromosoma de menor longitud.

[2] Un locus es una posición fija en un cromosoma, como la posición de un gen o de un marcador (marcador genético).

Estos operadores permiten, además, crear un *algoritmo genético de dos niveles*: a nivel de cromosoma y a nivel de gen. Supongamos que, en un problema de clasificación, hay un gen por clase. Se puede asignar una puntuación a cada gen en función del número de muestras que haya clasificado correctamente. Al aplicar estos operadores, se duplicarán los alelos[3] con mayor puntuación, y se eliminarán aquellos que hayan obtenido menor puntuación, o cuya puntuación sea nula.

Operadores de nicho (ecológico)

Otros operadores importantes son los operadores de *nicho*. Estos operadores están encaminados a mantener la diversidad genética de la población, de forma que cromosomas similares sustituyan sólo a cromosomas similares, y son especialmente útiles en problemas con muchas soluciones; un algoritmo genético con estos operadores es capaz de hallar todos los máximos, dedicándose cada especie a un máximo. Más que operadores genéticos, son formas de enfocar la selección y la evaluación de la población.

Operadores especializados

En una serie de problemas hay que restringir las nuevas soluciones generadas por los operadores genéticos, pues no todas las soluciones generadas van a ser válidas, sobre todo en los problemas con restricciones. Por ello, se aplican operadores que mantengan la estructura del problema. Otros operadores son simplemente generadores de diversidad, pero la generan de una forma determinada:

- **Zap**: en vez de cambiar un solo bit de un cromosoma, cambia un gen completo de un cromosoma.
- **Creep:** este operador aumenta o disminuye en 1 el valor de un gen; sirve para cambiar suavemente y de forma controlada los valores de los genes.
- **Transposición:** similar al crossover y a la recombinación genética, pero dentro de un solo cromosoma; dos genes intercambian sus valores, sin afectar al resto del cromosoma. Similar a este es el operador de **eliminación-reinserción**, en el que un gen cambia de posición con respecto a los demás.

Aplicando operadores genéticos

En toda ejecución de un algoritmo genético hay que decidir con qué frecuencia se va a aplicar cada uno de los algoritmos genéticos; en algunos casos, como en la mutación o el crossover uniforme, se debe de añadir algún parámetro adicional, que indique con qué frecuencia se va a aplicar dentro de cada gen del cromosoma. La frecuencia de aplicación de cada operador estará en función del problema; teniendo en cuenta los efectos de cada operador, tendrá que aplicarse con cierta frecuencia o no. Generalmente, la mutación y otros operadores que generen diversidad se suele aplicar con poca frecuencia; la recombinación se suele aplicar con frecuencia alta.

En general, la frecuencia de los operadores no varía durante la ejecución del algoritmo, pero hay que tener en cuenta que cada operador es más efectivo en un momento de la ejecución. Por ejemplo, *al principio, en la fase denominada de exploración, los más eficaces son la mutación y la recombinación;* posteriormente, cuando la población ha convergido en parte, la recombinación no es útil, pues se está trabajando con individuos bastante similares, y es poca la información que se intercambia. Sin embargo, si se produce un estancamiento, la mutación tampoco es útil, pues está reduciendo el algoritmo genético a una búsqueda aleatoria; y hay que aplicar otros operadores. En todo caso, se pueden usar operadores especializados.

[3] Un alelo o aleloide es cada una de las formas alternativas que puede tener un gen que se diferencian en su secuencia y que se puede manifestar en modificaciones concretas de la función de ese gen

Ventajas y Desventajas

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 las modernas arquitecturas masivamente paralelas.
- Usan operadores probabilísticos, en vez de los típicos operadores determinísticos de las otras técnicas.
- 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, etc.-.
- Pueden converger prematuramente debido a una serie de problemas de diversa índole.

Limitaciones

El poder de los Algoritmos Genéticos proviene del hecho de que se trata de una técnica robusta, y pueden tratar con éxito una gran variedad de problemas provenientes de diferentes áreas, incluyendo aquellos en los que otros métodos encuentran dificultades. Si bien no se garantiza que el Algoritmo Genético encuentre la solución óptima, del problema, existe evidencia empírica de que se encuentran soluciones de un nivel aceptable, en un tiempo competitivo con el resto de algoritmos de optimización combinatoria. En el caso de que existan técnicas especializadas para resolver un determinado problema, lo más probable es que superen al Algoritmo Genético, tanto en rapidez como en eficacia. *El gran campo de aplicación de los Algoritmos Genéticos se relaciona con aquellos problemas para los cuales no existen técnicas especializadas.* Incluso en el caso en que dichas técnicas existan, y funcionen bien, pueden efectuarse mejoras de las mismas hibridándolas con los Algoritmos Genéticos.

Como Saber si es Posible usar un Algoritmo Genético

La aplicación más común de los algoritmos genéticos ha sido la solución de problemas de optimización, en donde han mostrado ser muy eficientes y confiables. Sin embargo, no todos los problemas pudieran ser apropiados para la técnica, y se recomienda en general tomar en cuenta las siguientes características del mismo antes de intentar usarla:

- Su espacio de búsqueda (sus posibles soluciones) debe estar delimitado dentro de un cierto rango.
- Debe poderse definir una función de aptitud que nos indique qué tan buena o mala es una cierta respuesta.
- Las soluciones deben codificarse de una forma que resulte relativamente fácil de implementar en la computadora.

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

La **función de aptitud** no es más que la función objetivo de nuestro problema de optimización. El algoritmo genético únicamente maximiza, pero la minimización puede realizarse fácilmente utilizando el recíproco de la función maximizante (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 tiene 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 soluciones 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.

Marco de Desarrollo

El término Computación Evolutiva se refiere al estudio de los fundamentos y aplicaciones de ciertas técnicas heurísticas de búsqueda basadas en los principios naturales de la evolución.

Una gran variedad de algoritmos evolutivos han sido propuestos pero principalmente pueden clasificarse en:

- Algoritmos Genéticos,
- Programación Evolutiva,
- Estrategias Evolutivas,
- Sistemas Clasificadores y
- Programación Genética.

Esta clasificación se basa sobre todo en detalles de desarrollo histórico más que en el hecho de un funcionamiento realmente diferente, de hecho las bases biológicas en las que se apoyan son esencialmente las mismas.

Las diferencias entre ellos se centra en los operadores que se usan en cada caso y en general en la forma de implementar la selección, reproducción y sustitución de individuos en una población.

Aunque los detalles de la evolución no han sido completamente comprendidos, incluso hoy en día, existen algunos puntos en los que se fundamentan:

- La evolución es un proceso que opera a nivel de cromosomas, y no a nivel de individuos. Cada individuo es codificado como un conjunto de cromosomas.
- La selección natural es el mecanismo mediante el cual los individuos mejor adaptados son los que tienen mayores posibilidades de reproducirse.
- El proceso evolutivo tiene lugar en la etapa de la reproducción. Es en esta etapa donde se producen la mutación, que es la causante de que los cromosomas de los hijos puedan ser diferentes a los de los padres, y el cruce, que combina los cromosomas de los padres para que los hijos tengan cromosomas diferentes.

Los **Algoritmos Genéticos** resuelven los problemas generando poblaciones sucesivas a las que se aplican los operadores de mutación y cruce. Cada individuo representa una solución al problema, y se trata de encontrar al individuo que represente a la mejor solución.

La **Programación Genética** funciona igual que la técnica anterior pero se centra en el estudio de problemas cuya solución es un programa. De manera que los individuos de la población son programas que se acercan más o menos a realizar una tarea que es la solución.

La **Programación Evolutiva** es otro enfoque de los algoritmos genéticos, en este caso el estudio se centra en conseguir operadores genéticos que imiten lo mejor posible a la naturaleza, en cada caso, más que en la relación de los padres con su descendencia. En este caso no se utiliza el operador de cruce, tomando la máxima importancia el operador de mutación.

Estrategias Evolutivas se centran en el estudio de problemas de optimización e incluyen una visión del aprendizaje en dos niveles: a nivel de genotipo, y a nivel de fenotipo.

Los **Sistemas Clasificadores** engloban el estudio de problemas en los que la solución buscada se corresponde con toda una población.

Comparación con otros métodos de optimización

Algoritmos Genéticos y Matemáticos

Existen problemas de optimización que pueden ser resueltos por la implementación de un algoritmo tradicional. En este caso lo más conveniente es utilizarlo.

Por ejemplo para resolver un problema que requiera como solución saber solamente cual número es más grande, resulta más eficaz utilizar el algoritmo matemático directamente.

Sin embargo, éstos no son aplicables a problemas que posean algunas de estas características:

- La función representativa del problema no es continua. En este caso el mismo no es computable. Los algoritmos genéticos pueden trabajar con todo tipo de funciones ya que encontrarán un mínimo aceptable si no es posible encontrar el óptimo.
- La función representativa es dinámica: La relación entre las variable cambia dependiendo de los valores que tomen las mismas. Esta relación puede ser advertida o no. Las reglas del tipo

"X es igual a Y si el valor de X es chico;

X es 1.5 de y si el valor de X es grande

no se sabe que pasa para valores medios de X"

no pueden ser convertidas en un algoritmo algebraico ya que existen valores que se desconocen. A diferencia de un algoritmo tradicional, un algoritmo genético puede ser diseñado para trabajar bajo estas condiciones.

Algoritmos Genéticos y Métodos Enumerativos

Existe la posibilidad teórica de encontrar soluciones a problemas de optimización enumerando todas las soluciones posibles para todos los casos y posteriormente buscando la misma en la base de datos resultante. Los problemas se limitan entonces a un sistema de búsqueda eficiente del caso concreto. Por ejemplo los libros con tablas de logaritmos tradicionales constan de una larga serie de cálculos para todos los valores usuales. La solución consiste simplemente en buscar en la lista el número decimal y retornar el logaritmo dado.

La memorización de las tablas de multiplicar que se enseñan a los niños es otro ejemplo usual. Se espera que ante la pregunta ¿Cuánto es siete por cinco? los niños respondan instantáneamente "35" sin tener que estar calculando mentalmente la multiplicación.

Este método es factible siempre que el número de valores sea manejable. De otra manera el simple cálculo de los mismos se vuelve imposible. Ejemplo: Generar una tabla que contenga todas las movidas de todos los partidos posibles de un juego de damas resultaría imposible de hacer en la práctica.

La "memorización" de una serie de datos no es otra cosa que la construcción en la memoria del equivalente a una base de datos en donde se busca la pregunta y se encuentra automáticamente la respuesta.

Los algoritmos genéticos usan heurística para la resolución de problemas, lo cual limita drásticamente el número de datos a utilizar.

Algoritmos Genéticos y Sistemas Expertos

Un **Sistema Experto** es un programa de computadora que encuentra soluciones a problemas del tipo condicional con la estructura:

Si ocurren los hechos A,B,C,D , cual sería el valor del suceso E

Ejemplo: Si un análisis médico detecta los síntomas A , B , C y D en un paciente , ¿Cual será la enfermedad del sujeto?

Ejemplo: Si el análisis geológico de una capa de suelo detecta la presencia de los compuestos químicos A , B , C y D ¿Es factible que exista petróleo en la misma?

Si bien existen en la literatura ejemplos de la utilidad de ésta técnica, las reglas deben ser provistas por un especialista (o varios) en el tema. Por ende, se requiere que los conocimientos estén disponibles, que sean estructurados o factibles de ser estructurados (convertidos a reglas heurísticas) y que los hechos de la realidad sean relativamente estáticos, es decir que las causas para arribar a una determinada conclusión no cambien, ya que cada vez que esto sucede, los expertos deben reelaborar las reglas , lo cual dificulta y retarda considerablemente la operatoria del sistema.

Los Sistemas Expertos tuvieron su apogeo en la década de los 80`s , aproximadamente de 1979 a 1985. En esa época se los llegó a considerar verdaderas panaceas que resolverían muchos de los problemas cotidianos del hombre. Incluso se formaron en ese entonces varias compañías con el objeto específico de realizarlos y comercializarlos. Algunos fueron exitosos y funcionaron bien, pero las dificultades planteadas anteriormente no tardaron en aparecer. En particular:

- Existen temas en los cuales el conocimiento no es estático, sino que la aparición de nueva información altera las pautas o reglas de inferencia de los resultados. La necesidad permanentes de reevaluar las reglas por medio de expertos humanos lleva al sistema a una operatoria lenta y burocrática. Cada conocimiento nuevo implica reentrenar manualmente el sistema. Los Sistemas Expertos demostraron no ser útiles en este campo.
- Existen temas en los cuales la interrelación de ciertas variables no es conocida. Si la información disponible de cierto asunto es limitada, y no se conoce el comportamiento de algunas de sus variables, el Sistema experto tendrá grandes dificultades de programarse ya que sus reglas serán imprecisas.

El Cuadro siguiente muestra las condiciones básicas necesarias para la implementación efectiva de un sistema experto:

- Los expertos no siempre estructuran su conocimiento. Existen numerosas personas que razonan por métodos empíricos. Esto hace que les resulte muy difícil traducir sus pensamientos o su método deductivo a reglas que la computadora pueda interpretar. Un Sistema experto no podrá llegar a resultados valederos cuando los especialistas en un tema no puedan tener estructurados sus pensamientos. Por ejemplo: supóngase que se quiera programar un sistema experto para calificar obras de arte. Difícilmente se encontrará un crítico de arte que pueda estructurar las razones por las cuales considera "buena" o "mala" a una obra de arte. En general las palabras que pueda decir resultarán a los oídos del programador del Sistema como una serie de subjetividades imposibles de sistematizar.

Luego de observar todo esto, se empezó a considerar a los Sistemas expertos como aptos solamente para entornos reducidos y con condiciones de ejecución acotadas. La idea del Sistema Experto como " resolvedor universal de problemas " quedó sepultada.

Si bien la investigación básica de los algoritmos genéticos es contemporánea a la de los sistemas expertos, la renovada importancia que se les dio en el ámbito científico se produjo en paralelo a la desvalorización que sufrieron estos últimos.

Los algoritmos genéticos se revalorizaron ya que poseen las siguientes ventajas competitivas:

- Solo necesitan asesoramiento del experto cuando se agregan o suprimen variables al modelo. Los Sistemas Expertos requieren la presencia del mismo ante cada modificación del entorno.
- Los algoritmos genéticos solo requieren el asesoramiento del experto para identificar las variables pertinentes, aunque no es necesario que éstos definan sus valores ni sus relaciones (las reglas) iniciales o finales. Los Sistemas Expertos solo trabajan con las reglas y valores que les dictan los seres humanos.

Algoritmos Genéticos y Redes Neuronales

Una **red neuronal** es el intento de poder realizar una simulación computacional del comportamiento de partes del cerebro humano mediante la réplica en pequeña escala de los patrones que éste desempeña para la formación de resultados a partir de los sucesos percibidos. El cerebro consta de unidades llamadas neuronas, las cuales están conectadas entre si formando una red (de ahí la denominación "red neuronal")

Concretamente, se trata de poder analizar y reproducir el mecanismo de aprendizaje de sucesos que poseen los animales más evolucionados.

La red simula grupos de neuronas, llamados " capas " las cuales están relacionadas unas con otras. Los datos se introducen en la primera capa, llamada "capa de entradas" Cada capa transfiere la información a sus vecinas., teniendo un peso o ponderación para los valores, lo que va modificando los mismos en su paso a través de la red

Cuando los datos llegan a la última de las capas, llamada " capa de salida " el valor resultante es tomado como el resultado de la red. La red puede ser entrenada para diversos usos, entre ellos como mecanismo de optimización. En este sentido, se puede expresar que serían un modelo alternativo competitivo con los algoritmos genéticos, si se las programara para este fin.

En rigor de verdades, la literatura sugiere que se podrían hacer modelos mixtos o híbridos en donde se combinen las ventajas de las redes neuronales y los algoritmos genéticos, aunque hay muy poco material disponible en este campo. Tal vez esto se deba al hecho que los GA (Algoritmos Genéticos) y el estudio de las redes forman dos ramas o escuelas separadas dentro de la inteligencia artificial, por lo que existe una preferencia en los investigadores en perfeccionar alguno de los dos modelos antes que tratar de unirlos.

El Algoritmo Genético Simple

El Algoritmo Genético Simple, también denominado Canónico, se representa en la Figura . 1. Se necesita una codificación o representación del problema, que resulte adecuada al mismo. Además se requiere una función de ajuste ó adaptación al problema, la cual asigna un número real a cada posible solución codificada.

Durante la ejecución del algoritmo, los padres deben ser seleccionados para la reproducción, a continuación dichos padres seleccionados se cruzarán generando dos hijos, sobre cada uno de los cuales actuará un operador de mutación. El resultado de la combinación de las anteriores funciones será un conjunto de individuos (posibles soluciones al problema), los cuales en la evolución del Algoritmo Genético formarán parte de la siguiente población.

```
BEGIN /* Algoritmo Genetico Simple */
    Generar una poblacion inicial.
    Computar la funcion de evaluacion de cada individuo.
    WHILE NOT Terminado DO
    BEGIN /* Producir nueva generacion */
        FOR Tamaño poblacion/2 DO
        BEGIN /*Ciclo Reproductivo */
            Seleccionar dos individuos de la anterior generacion,
            para el cruce (probabilidad de seleccion proporcional
            a la funcion de evaluacion del individuo).
            Cruzar con cierta probabilidad los dos
            individuos obteniendo dos descendientes.
            Mutar los dos descendientes con cierta probabilidad.
            Computar la funcion de evaluacion de los dos
            descendientes mutados.
            Insertar los dos descendientes mutados en la nueva generacion.
        END
        IF la poblacion ha convergido THEN
            Terminado := TRUE
    END
END
```

Figura 1: Pseudocódigo del Algoritmo Genético Simple

Figura 1

Codificación

Se supone que los individuos (posibles soluciones del problema), pueden representarse como un conjunto de parámetros (que denominaremos genes), los cuales agrupados forman una ristra de valores (a menudo referida como cromosoma). Si bien el alfabeto utilizado para representar los individuos no debe necesariamente estar constituido por el (0, 1), buena parte de la teoría en la que se fundamentan los Algoritmos Genéticos utiliza dicho alfabeto.

En términos biológicos, el conjunto de parámetros representando un cromosoma particular se denomina fenotipo. El fenotipo contiene la información requerida para construir un organismo, el cual se refiere como genotipo. Los mismos términos se utilizan en el campo de los Algoritmos Genéticos.

La adaptación al problema de un individuo depende de la evaluación del genotipo. Esta última puede inferirse a partir del fenotipo, es decir puede ser computada a partir del cromosoma, usando la función de evaluación. La función de adaptación debe ser diseñada para cada problema de manera específica. Dado un cromosoma particular, la función de adaptación le asigna un número real, que se supone refleja el nivel de adaptación al problema del individuo representado por el cromosoma.

Durante la fase reproductiva se seleccionan los individuos de la población para cruzarse y producir descendientes, que constituirán, una vez. mutados, la siguiente generación de individuos.

La selección de padres se efectúa al azar usando un procedimiento que favorezca a los individuos mejor adaptados, ya que a cada individuo se le asigna una probabilidad de ser seleccionado que es proporcional a su función de adaptación. Este procedimiento se dice que está basado en la ruleta sesgada. Según dicho esquema, los individuos bien adaptados se escogerán probablemente varias veces por generación, mientras que, los pobremente adaptados al problema, no se escogerán más que de vez en cuando.

Una vez seleccionados dos padres, sus cromosomas se combinan, utilizando habitualmente los operadores de cruce y mutación.

El *operador de cruce*, toma dos padres seleccionados y corta sus ristras de cromosomas en una posición elegida al azar, para producir dos subristras iniciales y dos subristras finales. Después se intercambian las subristras finales, produciéndose dos nuevos cromosomas completos (representado en la Figura 2). Ambos descendientes heredan genes de cada uno de los padres. Este operador se conoce como operador de cruce basado en un punto.

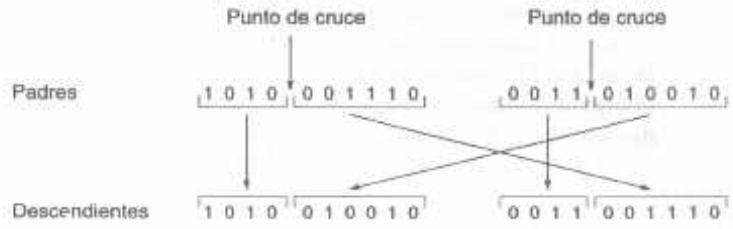


Figura 2: Operador de cruce basado en un punto

Habitualmente el operador de cruce no se aplica a todos los pares de individuos que han sido seleccionados para emparejarse, sino que se aplica de manera aleatoria, normalmente con una probabilidad comprendida entre 0.5 y 1.0. En el caso en que el operador de cruce no se aplique, la descendencia se obtiene simplemente duplicando los padres.

El *operador de mutación* se aplica a cada hijo de manera individual, y consiste en la alteración aleatoria (normalmente con probabilidad pequeña) de cada gen componente del cromosoma. Se muestra la mutación del quinto gen del cromosoma. Sí bien puede en principio pensarse que el operador de cruce es más importante que el operador de mutación, ya que proporciona una exploración rápida del espacio de búsqueda, éste último asegura que ningún punto del espacio de búsqueda tenga probabilidad cero de ser examinado, y es de capital importancia para asegurar la convergencia de los Algoritmos Genéticos.

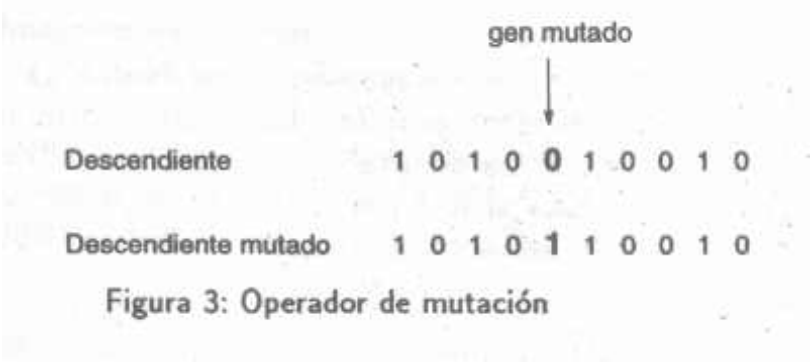


Figura 3: Operador de mutación

Para criterios prácticos, es muy útil la definición de convergencia introducida en este campo por De Jong. **Si el Algoritmo Genético ha sido correctamente implementado, la población evolucionará a lo largo de las generaciones sucesivas de tal manera que la adaptación media extendida a todos los individuos de la población, así como la adaptación del mejor individuo se irán incrementando hacia el óptimo global.**

El concepto de convergencia está relacionado con la progresión hacia la uniformidad: un gen ha convergido cuando al menos el 95 % de los individuos de la población comparten el mismo

valor para dicho gen. Se dice que la población converge cuando todos los genes han convergido. Se puede generalizar dicha definición al caso en que al menos un poco de los individuos de la población hayan convergido.

Se muestra como varía la adaptación media y la mejor adaptación en un Algoritmo Genético Simple típico.

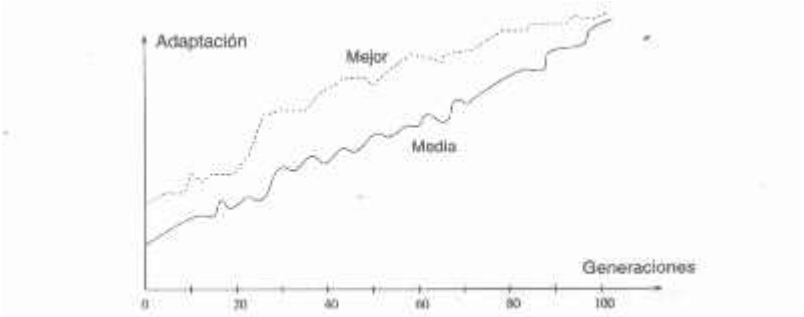


Figura 4: Adaptación media y mejor adaptación en un Algoritmo Genético Simple

A medida que el número de generaciones aumenta, es más probable que la adaptación media se aproxime a la del mejor individuo.

Ejemplo: Resolución de un algoritmo genético de manera manual

Aplicaremos AG paso a paso, para aclarar su funcionamiento, a un simple problema de optimización.

Consideremos encontrar el máximo de la función $f(x)=x^2$ definida sobre el conjunto de enteros entre 0 y 31.
El espacio solución es {x/x pertenece a Z y 0<=x<=31}. Para representar estos elementos como tira, basta considerar a los enteros en su representación binaria.

El espacio solución puede verse entonces como {00000, 00001, 00010, 00011,…, 11111}. Es decir que el espacio solución esta formado por cromosomas de longitud 5 y cada gen es un dígito binario 0 o 1.

Nuestro próximo paso consiste en crear la población inicial con cromosomas elegidos al azar. La cantidad de miembros de esta población, dado lo reducido del tamaño del espacio solución, la estableceremos en 4.

Elegimos cuatro cromosomas al azar.
Supongamos que los cuatro seleccionados son 01101, 11000, 01000 y 10011.

Pobl 1	X	F obj.	Fitness
1 01101	13	169	0.14
2 11000	24	576	0.49
3 01000	8	64	0.06
4 10011	19	361	0.31
suma		1170	1
promedio		293	0.25
máximo		576	0.49

Vayamos reflejando los cálculos y anotaciones a medida que avancemos en la tabla que se encuentra a la izquierda con nombre Pobl1.

La primera columna es el número de orden de cada tira.
La segunda, menciona el cromosoma correspondiente.
La tercera, el valor entero del cromosoma.
La cuarta, el valor de la función objetivo correspondiente a ese cromosoma.
En la quinta, el fitness del cromosoma, evaluado como el cociente entre el valor de la función sobre la suma de todos los valores funcionales. (F.Obj / Σ F.Obj)
Obtenemos respectivamente 0.14, 0.49, 0.06 y 0.31.

Debajo de ellos calculamos la suma, el promedio y el máximo.

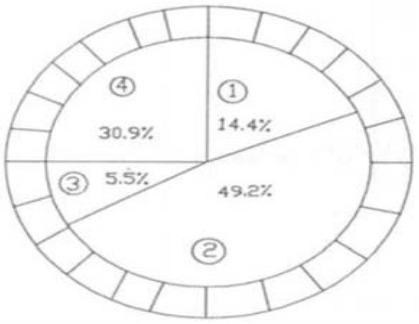
Al comenzar este proceso debieron suministrarse dos valores la probabilidad de crossover (PC=0.90) y la probabilidad de mutación (PM=0.001).

Entonces, comenzamos a trabajar para la obtención de la siguiente población.

Pobl 2	X	F obj.	Fitness
1 01100	12	144	
2 11001	25	625	
3 11000	27	729	
4 10011	16	256	
suma		1754	
promedio		439	
máximo		729	

La primera fase es la selección, o sea la extracción de cromosomas que jugarán el papel de padre con el objeto de obtener hijos para la nueva población. Para ello usaremos, como operador genético, el método de la ruleta.

Para construirla, a cada cromosoma se le asigna un arco de circunferencia proporcional a su fitness.



Giramos, entonces, la ruleta 4 veces obteniendo los siguientes resultados: 1, 2, 2, 4.
Se toma el primer par (1,2), se consulta la probabilidad de crossover. Supongamos que en este caso la respuesta es SÍ.

Para realizar el crossover, usaremos el operador genético "crossover de un corte". Consiste en elegir al azar el lugar en donde se cortarán ambos cromosomas padres: supongamos 4. Entonces, ambos padres ceden los genes del 1 al 4, a los hijos, mientras que el quinto gen se intercambia. Es decir: el primer hijo está formado por los 4 primeros genes del primer padre y el quinto gen del segundo padre, mientras que el segundo hijo está formado por los 4 primeros genes del segundo padre y el quinto gen del primer padre. O sea, los hijos gestados son 01100 y 11001.

Se toma, ahora, el segundo par (2,4). Supongamos que la probabilidad también es SÍ, y el corte se da en 2. Resultan, entonces, los siguientes hijos: 11000 y 10011 (ver Tabla. Pobl2)

En cuanto a la probabilidad de mutación, asumimos que la respuesta en cada uno de los 4 casos es NO. Si la respuesta hubiese sido afirmativa para alguno de los cromosomas, deberíamos haber cambiado el valor de un gen elegido al azar por otro mutado (en este caso 0 por 1 y 1 por 0)
En la tabla Pobl2 queda reflejada la nueva población.

La idea de este ejemplo es apreciar como mejora la calidad genética de la población 2 con respecto a la 1. Basta comparar los nuevos valores de las sumas, del promedio y el máximo.

Todo hace suponer que, continuando con este ciclo, las mejoras irían incrementándose paulatinamente.

En el ejercicio que se propone a continuación se pide graficar las curvas que manifiesten estos totales parciales.

Ejercicio

Hacer un programa que utilice un Algoritmo Genético Canónico para buscar un máximo de la función:

$f(x) = (x/coef)^2$ en el dominio $[0 , 2^{30} -1]$

donde $coef = 2^{30} -1$

teniendo en cuenta los siguientes datos:

Probabilidad de Crossover = 0,75
Probabilidad de Mutación = 0,05
Población Inicial: 10 individuos
Ciclos del programa: 20
Método de Selección: Ruleta
Método de Crossover: 1 Punto
Método de Mutación: invertida

El programa debe mostrar, finalmente, el Cromosoma correspondiente al valor máximo obtenido y gráficas, usando EXCEL, de Máx, Mín y Promedio de la función objetivo por cada generación

Índice de Contenidos:

Bibliografía:

"Informática evolutiva: Algoritmos genéticos". Juan Julián Merelo Guervós

Material de la cátedra de Algoritmos Genéticos. Ing. Néstor Berlande.

"Algoritmos Genéticos". Univ. San Carlos de Guatemala- William Orlando Machán Morente. On Line en: http://www.youtube.com/watch?v=qRehZZckG0I

"Algoritmos Genéticos". On Line en: http://eddyalfaro.galeon.com/geneticos.html

"Algoritmos Genéticos". Univ. Calors III- Jorge Arranz de la Peña- Antonio Parra Truyol. http://www.it.uc3m.es/jvillena/irc/practicas/06-07/05.pdf