Introducción a los Algoritmos Genéticos

Marcos Gestal Pose
Depto. Tecnologías de la Información y las Comunicaciones
Universidade da Coruña
http://sabia.tic.udc.es/~mgestal
mgestal@udc.es

Índice

Índice de figuras

El siguiente tutorial no pretende ser un documento exhaustivo acerca de los Algoritmos Genéticos. Más bien una referencia que sirva para introducir la terminología, los conceptos claves y una bibliografía de base, quedando en manos del lector profundizar en aquellos aspectos que considere de mayor interés.

1. Introducción

Los Algoritmos Genéticos son métodos adaptativos, generalmente usados en problemas de búsqueda y optimización de parámetros, basados en la reproducción sexual y en el principio supervivencia del más apto.

Más formalmente, y siguiendo la definición dada por Goldberg, *"los Algoritmos Genéticos son algoritmos de búsqueda basados en la mecánica de selección natural y de la genética natural. Combinan 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].

Para alcanzar la solución a un problema se parte de un conjunto inicial de individuos, llamado población, generado de manera aleatoria. Cada uno de estos individuos representa una posible solución al problema. Estos individuos evolucionarán tomando como base los esquemas propuestos por Darwin [Darwin, 1859] sobre la selección natural, y se adaptarán en mayor medida tras el paso de cada generación a la solución requerida.

2. Orígenes

Si algo funciona bien, ¿por qué no imitarlo?. La respuesta a esta pregunta nos lleva directamente a los orígenes de la computación evolutiva. Durante millones de años las diferentes especies se han adaptado para poder sobrevivir en un medio cambiante. De la misma manera se podría tener una población de potenciales soluciones a un problema de las que se irían seleccionando las mejores hasta que se adaptasen perfectamente al medio, en este caso el problema a resolver. En términos muy generales se podría definir la computación evolutiva como una familia de modelos computacionales inspirados en la evolución.

Más formalmente el término de computación evolutiva se refiere al estudio de los fundamentos y aplicaciones de ciertas técnicas heurísticas basadas en los principios de la evolución natural [Tomassini, 1995]. Estas técnicas heurísticas podrían clasificarse en 3 categorías principales dando lugar a la ecuación evolutiva (Fig. 1).

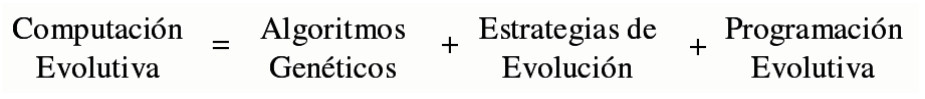

Figura 1: Ecuación Evolutiva

A continuación se detallarán un poco más los orígenes de cada una de las disciplinas participantes en dicha ecuación.

El desarrollo de los Algoritmos Genéticos se debe en gran medida a John Holland, investigador de la Universidad de Michigan. A finales de la década de los 60 desarrolló una técnica que imitaba en su funcionamiento a la selección natural. Aunque originalmente esta técnica recibió el nombre de *planes reproductivos*, a raíz de la publicación en 1975 de su libro "Adaptation in Natural and Artificial Systems" [Holland, 1975] se conoce principalmente con el nombre de Algoritmos Genéticos.

A grandes rasgos un Algoritmo Genético consiste en una población de soluciones codificadas de forma similar a cromosomas. Cada uno de estos cromosomas tendrá asociado un ajuste, valor de bondad, ajuste o fitness, que cuantifica su validez como solución al problema. En función de este valor se le darán más o menos oportunidades de reproducción. Además, con cierta probabilidad se realizarán mutaciones de estos cromosomas.

Las bases de las Estrategias de Evolución fueron apuntadas en 1973 por Rechemberg en su obra "Evolutionsstrategie: Optimierung Technisher Systeme nach Prinzipien der Biologischen Evolution" [Rechenberg, 1973]. Las dos Estrategias de Evolución más empleadas son la $(\mu + \lambda)$-ES y la (μ, λ)-ES. En la primera de ellas un total de μ producen λ descendientes reduciéndose nuevamente la población a μ individuos (los padres de la siguiente generación) por selección de los mejores individuos. De esta manera los padres sobreviven hasta que son reemplazados por hijos mejores que ellos. En la $(\mu+\lambda)$-ES la descendencia reemplaza directamente a los padres, sin hacer ningún tipo de comprobación.

La Programación Evolutiva surge principalmente a raíz del trabajo "Artificial Intelligence Through Simulated Evolution" de Fogel, Owens y Walsh, publicado en 1966 [Fogel et al., 1966]. En este caso los individuos, conocidos aquí como organismos, son máquinas de estado finito. Los organismos que mejor resuelven alguna de las funciones objetivo obtienen la oportunidad de reproducirse. Antes de producirse los cruces para generar la descendencia se realiza una mutación sobre los padres.

A su vez la computación evolutiva puede verse como uno de los campos de investigación de lo que se ha dado en llamar Soft Computing (Fig. 2).

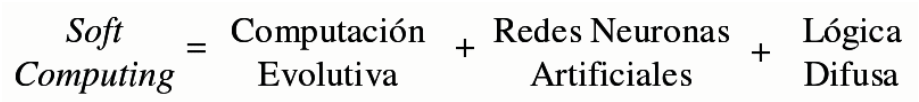

Figura 2: Soft Computing

Como se ha comentado anteriormente la computación evolutiva tiene una fuerte base biológica. En sus orígenes los algoritmos evolutivos consistieron en copiar procesos que tienen lugar en la selección natural. Este último concepto había sido introducido, rodeado de mucha polémica, por Charles Darwin [Darwin, 1859]. A pesar de que aún hoy en día no todos los detalles de la evolución biológica son completamente conocidos, existen algunos hechos apoyados sobre una fuerte evidencia experimental:

- La evolución es un proceso que opera, más que sobre los propios organismos, sobre los cromosomas. Estos cromosomas pueden ser considerados como herramientas orgánicas que codifican la vida, o visto al revés, una

criatura es *creada* decodificando la información contenida en los cromosomas.

- La selección natural es el mecanismo que relaciona los cromosomas con la eficiencia respecto al medio de la entidad que representan. Otorga a los individuos más adaptados al medio un mayor número de oportunidades de reproducirse.
- Los procesos evolutivos tienen lugar durante la etapa de reproducción. Aunque existe una larga serie de mecanismos que afectan a la reproducción los más comunes son la mutación, causante de que los cromosomas de la descendencia sean diferentes a los de los padres, y el cruce o recombinación, que combina los cromosomas de los padres para producir la descendencia.

3. Bases Biológicas

En la naturaleza, los individuos de una población compiten constantemente con otros por recursos tales como comida, agua y refugio. Los individuos que tienen más éxito en la lucha por los recursos tienen mayores probabilidades de sobrevivir y generalmente una descendencia mayor. Al contrario, los individuos peor adaptados tienen un menor número de descendientes, o incluso ninguno. Esto implica que los genes de los individuos mejor adaptados se propagarán a un número cada vez mayor de individuos de las sucesivas generaciones.

La combinación de características buenas de diferentes ancestros puede originar en ocasiones que la descendencia esté incluso mejor adaptada al medio que los padres. De esta manera, las especies evolucionan adaptándose más y más al medio a medida que transcurren las generaciones [Beasley et al., 1993].

Pero la adaptación de un individuo al medio no sólo está determinada por su composición genética. Influyen otros factores como el aprendizaje, en ocasiones adquirido por el método de prueba y error, en ocasiones adquirido por imitación del comportamiento de los padres. Para imitar esta adquisición de conocimiento se han desarrollado variantes como el Ajuste Fino.

Pero la adaptación de un individuo al medio no sólo está determinada por su composición genética. Influyen otros factores como el aprendizaje, en ocasiones adquirido por el método de prueba y error, en ocasiones adquirido por imitación del comportamiento de los padres. Para imitar esta adquisición de conocimiento pueden emplearse técnicas de Ajuste Fino, consistentes en pequeñas modificaciones de los genes de un cromosoma. Por ejemplo, estas modificaciones pueden realizarse tomando como resultado el mejor individuo tras la ejecución de varias generaciones de un Algoritmo Genético cuya población previamente ha sido creada a partir de ligeras variaciones del individuo al que se desea incorporar el conocimiento.

4. Codificación de Problemas

Cualquier solución potencial a un problema puede ser presentada dando valores a una serie de parámetros. El conjunto de todos los parámetros (genes en la terminología de Algoritmos Genéticos) se codifica en una cadena de valores denominada cromosoma.

El conjunto de los parámetros representado por un cromosoma particular recibe el nombre de genotipo. El genotipo contiene la información necesaria para la construcción del organismo, es decir, la solución real al problema, denominada fenotipo. Por ejemplo, en términos biológicos, la información genética contenida en el ADN de un individuo sería el genotipo, mientras que la expresión de ese ADN (el propio individuo) sería el fenotipo.

Desde los primeros trabajos de John Holland la codificación suele hacerse mediante valores binarios. Se asigna un determinado número de bits a cada parámetro y se realiza una discretización de la variable representada por cada gen. El número de bits asignados dependerá del grado de ajuste que se desee alcanzar. Evidentemente no todos los parámetros tienen porque estar codificados con el mismo número de bits. Cada uno de los bits pertenecientes a un gen suele recibir el nombre de alelo.

La figura 3 muestra un ejemplo de un individuo binario que codifica 3 parámetros.

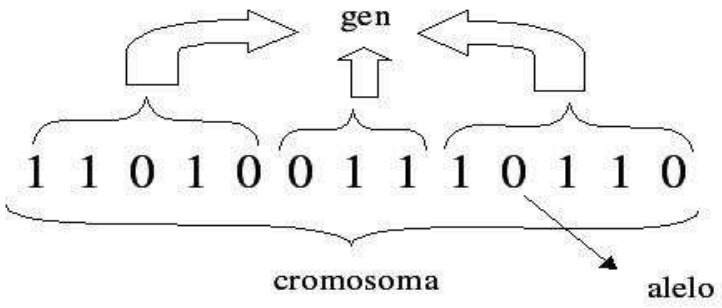

Figura 3: Individuo Genético Binario

Sin embargo, también pueden existir representaciones que codifiquen directamente cada parámetro con un valor entero, real o en punto flotante. A pesar de que se acusa a estas representaciones de degradar el paralelismo implícito de las representaciones binarias, permiten el desarrollo de operadores genéticos más específicos al campo de aplicación del Algoritmo Genético.

Un ejemplo típico de la aplicación de los Algoritmos Genéticos es la optimización de los pesos de una red de neuronas artificiales. La codificación de la red en forma de cromosoma (fig. 4) es tan sencilla como asignar un gen del cromosoma a cada uno de los pesos de la red. También se podrían añadir genes que indicasen el número de capas de la red, y el número de elementos de procesado en cada capa.

5. Algoritmo Principal

Los Algoritmos Genéticos trabajan sobre una población de individuos. Cada uno de ellos representa una posible solución al problema que se desea resolver. Todo individuo tiene asociado un ajuste de acuerdo a la bondad con respecto al problema de la solución que representa (en la naturaleza el equivalente sería una medida de la eficiencia del individuo en la lucha por los recursos).

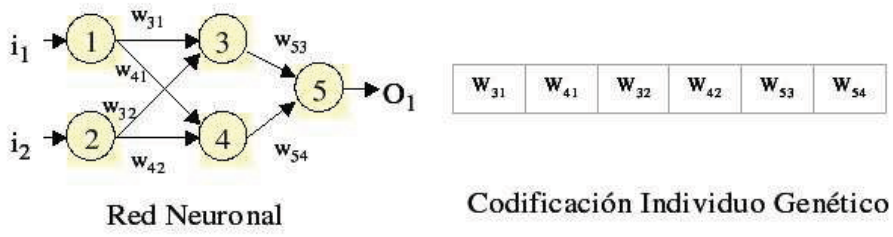

Figura 4: Ejemplo Codificación: Red de Neuronas Artificiales

El funcionamiento genérico de un Algoritmo Genético puede apreciarse en el siguiente pseudocódigo:

```
Inicializar población actual aleatoriamente

MIENTRAS no se cumpla el criterio de terminación
       crear población temporal vacía

  MIENTRAS  población temporal no llena
        seleccionar padres
        cruzar padres con probabilidad Pc
        SI se ha producido el cruce
            mutar uno de los descendientes con probabilidad Pm
            evaluar descendientes
            añadir descendientes a la población temporal
        SINO
            añadir padres a la población temporal
        FIN SI
  FIN MIENTRAS

  aumentar contador generaciones
  establecer como nueva población actual la población temporal

FIN MIENTRAS
```

Una generación se obtiene a partir de la anterior por medio de los operadores de reproducción. Existen 2 tipos:

- **Cruce:** Se trata de una reproducción de tipo sexual. Se genera una descendencia a partir del mismo número de individuos (generalmente 2) de la generación anterior. Existen varios tipos que se detallarán en un punto posterior.
- **Copia:** Se trata de una reproducción de tipo asexual. Un determinado número de individuos pasa sin sufrir ninguna variación directamente a la siguiente generación.

Una vez generados los nuevos individuos se realiza la mutación con una probabilidad Pm. La probabilidad de mutación suele ser muy baja, por lo general entre el 0.5 % y el 2 %.

Se sale de este proceso cuando se alcanza alguno de los criterios de parada fijados. Los más usuales suelen ser:

- Los mejores individuos de la población representan soluciones suficientemente buenas para el problema que se desea resolver.
- La población ha convergido. Un gen ha convergido cuando el 95 % de la población tiene el mismo valor para él, en el caso de trabajar con codificaciones binarias, o valores dentro de un rango especificado, en el caso de trabajar con otro tipo de codificaciones. Una vez que todos los genes alcanzan la convergencia se dice que la población ha convergido. Cuando esto ocurre la media de bondad de la población se aproxima a la bondad del mejor individuo.
- Se ha alcanzado el número de generaciones máximo especificado.

Sobre este algoritmo inicialmente propuesto por Holland se han definido numerosas variantes.

Quizás una de las más extendidas consiste en prescindir de la población temporal de manera que los operadores genéticos de cruce y mutación se aplican directamente sobre la población genética. Con esta variante el proceso de cruces varía ligeramente. Ahora no basta, en el caso de que el cruce se produzca, con insertar directamente la descendencia en la población. Puesto que el número de individuos de la población se ha de mantener constante, antes de insertar la descendencia en la población se le ha de hacer sitio. Existen para ello diversas opciones:

- **Reemplazo de padres:** para hacer hueco a la descendencia en la población se eliminan de ella a los padres.
- **Reemplazo de individuos similares:** cada uno de los individuos de la descendencia reemplazará a un individuo de la población con un ajuste similar al suyo. Para escoger este individuo se obtiene la posición en la que se debería insertar el nuevo individuo para mantener ordenada la población y se escoge para insertarlo una posición al azar de su vecindad (p.e. uno de entre los cinco individuos superiores o inferiores).
- **Reemplazo de los peores individuos:** los individuos que se eliminarán de la población para dejar paso a la descendencia se seleccionarán aleatoriamente de entre los peores individuos de la población. Por lo general se consideran individuos pertenecientes al último 10 %.
- **Reemplazo aleatorio:** los individuos eliminados se seleccionan al azar.

Evidentemente trabajando con un única población no se puede decir que se pase a la siguiente generación cuando se llene la población, pues siempre está llena. En este caso el paso a la siguiente generación se producirá una vez que se hayan alcanzado cierto número de cruces y mutaciones. Este número dependerá de la tasa de cruces y mutaciones especificadas por el usuario y

del tamaño de la población. Así con una tasa de cruces del 90 %, una tasa de mutaciones del 0.02 % y trabajando con 100 individuos se pasará a la siguiente generación cuando se alcanzasen 45 cruces (cada cruce genera 2 individuos con lo que se habrían insertado en la población 90 individuos, esto es el 90 %) o 2 mutaciones.

6. Operadores Genéticos

Para el paso de una generación a la siguiente se aplican una serie de operadores genéticos. Los más empleados son los operadores de selección, cruce, copia y mutación. En el caso de no trabajar con una población intermedia temporal también cobran relevancia los algoritmos de reemplazo.

A continuación se verán en mayor detalle.

6.1. Selección

Los algoritmos de selección serán los encargados de escoger qué individuos van a disponer de oportunidades de reproducirse y cuáles no.

Puesto que se trata de imitar lo que ocurre en la naturaleza, se ha de otorgar un mayor número de oportunidades de reproducción a los individuos más aptos. Por lo tanto la selección de un individuo estará relacionada con su valor de ajuste. No se debe sin embargo eliminar por completo las opciones de reproducción de los individuos menos aptos, pues en pocas generaciones la población se volvería homogénea.

Una opción bastante común consiste en seleccionar el primero de los individuos participantes en el cruce mediante alguno de los métodos expuestos a continuación y el segundo de manera aleatoria.

6.1.1. Selección por ruleta

Propuesto por DeJong, es posiblemente el método más utilizado desde los orígenes de los Algoritmos Genéticos [Blickle and Thiele, 1995].

A cada uno de los individuos de la población se le asigna una parte proporcional a su ajuste de una ruleta, de tal forma que la suma de todos los porcentajes sea la unidad. Los mejores individuos recibirán una porción de la ruleta mayor que la recibida por los peores. Generalmente la población está ordenada en base al ajuste por lo que las porciones más grandes se encuentran al inicio de la ruleta. Para seleccionar un individuo basta con generar un número aleatorio del intervalo [0..1] y devolver el individuo situado en esa posición de la ruleta. Esta posición se suele obtener recorriendo los individuos de la población y acumulando sus proporciones de ruleta hasta que la suma exceda el valor obtenido.

Es un método muy sencillo, pero ineficiente a medida que aumenta el tamaño de la población (su complejidad es $O(n^2)$). Presenta además el inconveniente de que el peor individuo puede ser seleccionado más de una vez.

En mucha bibliografía se suele referenciar a este método con el nombre de Selección de Montecarlo.

6.1.2. Selección por torneo

La idea principal de este método consiste en realizar la selección en base a comparaciones directas entre individuos. Existen dos versiones de selección mediante torneo:

- Determinística
- Probabilística

En la versión determinística se selecciona al azar un número p de individuos (generalmente se escoge $p = 2$). De entre los individuos seleccionados se selecciona el más apto para pasarlo a la siguiente generación.

La versión probabilística únicamente se diferencia en el paso de selección del ganador del torneo. En vez de escoger siempre el mejor se genera un número aleatorio del intervalo [0..1], si es mayor que un parámetro p (fijado para todo el proceso evolutivo) se escoge el individuo más alto y en caso contrario el menos apto. Generalmente p toma valores en el rango $0{,}5 < p \leq 1$.

Variando el número de individuos que participan en cada torneo se puede modificar la presión de selección. Cuando participan muchos individuos en cada torneo, la presión de selección es elevada y los peores individuos apenas tienen oportunidades de reproducción. Un caso particular es el *elitismo global*. Se trata de un torneo en el que participan todos los individuos de la población con lo cual la selección se vuelve totalmente determinística. Cuando el tamaño del torneo es reducido, la presión de selección disminuye y los peores individuos tienen más oportunidades de ser seleccionados.

Elegir uno u otro método de selección determinará la estrategia de búsqueda del Algoritmo Genético. Si se opta por un método con una alta presión de selección se centra la búsqueda de las soluciones en un entorno próximo a las mejores soluciones actuales. Por el contrario, optando por una presión de selección menor se deja el camino abierto para la exploración de nuevas regiones del espacio de búsqueda.

Existen muchos otros algoritmos de selección. Unos buscan mejorar la eficiencia computacional, otros el número de veces que los mejores o peores individuos pueden ser seleccionados. Algunos de estos algoritmos son muestreo determinístico, escalamiento sigma, selección por jerarquías, estado uniforme, sobrante estocástico, brecha generacional, etc.

6.2. Cruce

Una vez seleccionados los individuos, éstos son recombinados para producir la descendencia que se insertará en la siguiente generación. Tal y como se ha indicado anteriormente el cruce es una estrategia de reproducción sexual.

Su importancia para la transición entre generaciones es elevada puesto que las tasas de cruce con las que se suele trabajar rondan el 90 %.

Los diferentes métodos de cruce podrán operar de dos formas diferentes. Si se opta por una estrategia destructiva los descendientes se insertarán en la población temporal aunque sus padres tengan mejor ajuste (trabajando con una única población esta comparación se realizará con los individuos a reemplazar). Por el contrario utilizando una estrategia no destructiva la descendencia pasará a la siguiente generación únicamente si supera la bondad del ajuste de los padres

(o de los individuos a reemplazar). La idea principal del cruce se basa en que, si se toman dos individuos correctamente adaptados al medio y se obtiene una descendencia que comparta genes de ambos, existe la posibilidad de que los genes heredados sean precisamente los causantes de la bondad de los padres. Al compartir las características buenas de dos individuos, la descendencia, o al menos parte de ella, debería tener una bondad mayor que cada uno de los padres por separado. Si el cruce no agrupa las mejores características en uno de los hijos y la descendencia tiene un peor ajuste que los padres no significa que se esté dando un paso atrás. Optando por una estrategia de cruce no destructiva garantizamos que pasen a la siguiente generación los mejores individuos. Si, aún con un ajuste peor, se opta por insertar a la descendencia, y puesto que los genes de los padres continuarán en la población - aunque dispersos y posiblemente levemente modificados por la mutación - en posteriores cruces se podrán volver a obtener estos padres, recuperando así la bondad previamente perdida.

Existen multitud de algoritmos de cruce. Sin embargo los más empleados son los que se detallarán a continuación:

- Cruce de 1 punto
- Cruce de 2 puntos
- Cruce uniforme

6.2.1. Cruce de 1 punto

Es la más sencilla de las técnicas de cruce. Una vez seleccionados dos individuos se cortan sus cromosomas por un punto seleccionado aleatoriamente para generar dos segmentos diferenciados en cada uno de ellos: la cabeza y la cola. Se intercambian las colas entre los dos individuos para generar los nuevos descendientes. De esta manera ambos descendientes heredan información genética de los padres, tal y como puede verse en la figura 5.

En la bibliografía suele referirse a este tipo de cruce con el nombre de SPX (Single Point Crossover)

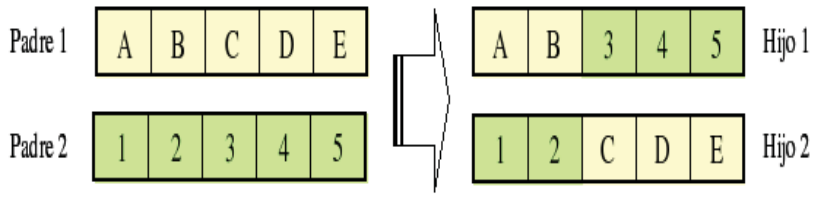

Figura 5: Cruce de 1 Punto

6.2.2. Cruce de 2 puntos

Se trata de una generalización del cruce de 1 punto. En vez de cortar por un único punto los cromosomas de los padres como en el caso anterior se realizan dos cortes. Deberá tenerse en cuenta que ninguno de estos puntos de corte coincida con el extremo de los cromosomas para garantizar que se originen tres

segmentos. Para generar la descendencia se escoge el segmento central de uno de los padres y los segmentos laterales del otro padre (ver figura 6.

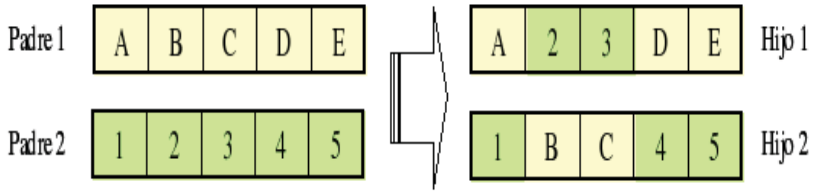

Figura 6: Cruce de 2 Puntos

Generalmente se suele referir a este tipo de cruce con las siglas DPX (Double Point Crossover).

Generalizando se pueden añadir más puntos de cruce dando lugar a algoritmos de cruce multipunto. Sin embargo existen estudios que desaprueban esta técnica [Jong, 1975] . Aunque se admite que el cruce de 2 puntos aporta una sustancial mejora con respecto al cruce de un solo punto, el hecho de añadir un mayor número de puntos de cruce reduce el rendimiento del Algoritmo Genético. El problema principal de añadir nuevos puntos de cruce radica en que es más fácil que los segmentos originados sean corrompibles, es decir, que por separado quizás pierdan las características de bondad que poseían conjuntamente. Sin embargo no todo son desventajas y añadiendo más puntos de cruce se consigue que el espacio de búsqueda del problema sea explorado más a fondo.

6.2.3. Cruce Uniforme

El cruce uniforme es una técnica completamente diferente de las vistas hasta el momento. Cada gen de la descendencia tiene las mismas probabilidades de pertenecer a uno u otro padre.

Aunque se puede implementar de muy diversas formas, la técnica implica la generación de una máscara de cruce con valores binarios. Si en una de las posiciones de la máscara hay un 1, el gen situado en esa posición en uno de los descendientes se copia del primer padre. Si por el contrario hay un 0 el gen se copia del segundo padre. Para producir el segundo descendiente se intercambian los papeles de los padres, o bien se intercambia la interpretación de los unos y los ceros de la máscara de cruce.

Tal y como se puede apreciar en la figura 7, la descendencia contiene una mezcla de genes de cada uno de los padres. El número efectivo de puntos de cruce es fijo pero será por término medio $L/2$, siendo L la longitud del cromosoma (número de alelos en representaciones binarias o de genes en otro tipo de representaciones).

Se suele referir a este tipo de cruce con las siglas UPX (Uniform Point Crossover).

6.2.4. Cruces específicos de codificaciones no binarias

Los tres tipos de cruce vistos hasta el momento son válidos para cualquier tipo de representación del genotipo. Si se emplean genotipos compuestos por

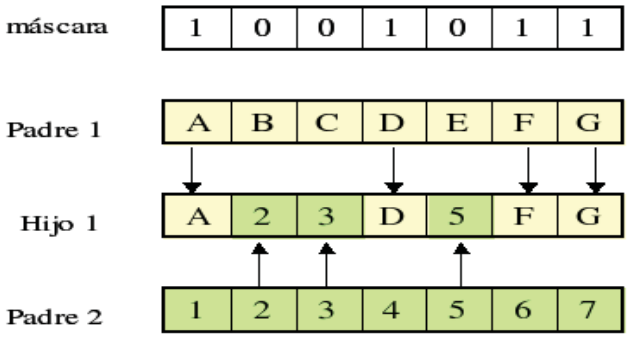

Figura 7: Cruce Uniforme

valores enteros o reales pueden definirse otro tipo de operadores de cruce:

- **Media:** el gen de la descendencia toma el valor medio de los genes de los padres. Tiene la desventaja de que únicamente se genera un descendiente en el cruce de dos padres.
- **Media geométrica:** cada gen de la descendencia toma como valor la raíz cuadrada del producto de los genes de los padres. Presenta el problema añadido de qué signo dar al resultado si los padres tienen signos diferentes.
- **Extensión**: se toma la diferencia existente entre los genes situados en las mismas posiciones de los padres y se suma al valor más alto o se resta del valor más bajo. Solventa el problema de generar un único descendiente.

6.3. Algoritmos de Reemplazo

Cuando en vez de trabajar con una población temporal se hace con una única población, sobre la que se realizan las selecciones e inserciones, deberá tenerse en cuenta que para insertar un nuevo individuo deberá de eliminarse previamente otro de la población. Existen diferentes métodos de reemplazo:

- **Aleatorio:** el nuevo individuo se inserta en un lugar cualquiera de la población.
- **Reemplazo de padres:** se obtiene espacio para la nueva descendencia liberando el espacio ocupado por los padres.
- **Reemplazo de similares:** una vez obtenido el ajuste de la descendencia se selecciona un grupo de individuos (entre seis y diez) de la población con un ajuste similar. Se reemplazan aleatoriamente los que sean necesarios.
- **Reemplazo de los peores:** de entre un porcentaje de los peores individuos de la población se seleccionan aleatoriamente los necesarios para dejar sitio a la descendencia.

6.4. Copia

La copia es la otra estrategia reproductiva para la obtención de una nueva generación a partir de la anterior. A diferencia del cruce, se trata de una estrategia de reproducción asexual. Consiste simplemente en la copia de un individuo en la nueva generación.

El porcentaje de copias de una generación a la siguiente es relativamente reducido, pues en caso contrario se corre el riesgo de una convergencia prematura de la población hacia ese individuo. De esta manera el tamaño efectivo de la población se reduciría notablemente y la búsqueda en el espacio del problema se focalizaría en el entorno de ese individuo.

Lo que generalmente se suele hacer es seleccionar dos individuos para el cruce, y si éste finalmente no tiene lugar, se insertan en la siguiente generación los individuos seleccionados.

6.5. Mutación

La mutación de un individuo provoca que alguno de sus genes, generalmente uno sólo, varíe su valor de forma aleatoria.

Aunque se pueden seleccionar los individuos directamente de la población actual y mutarlos antes de introducirlos en la nueva población, la mutación se suele utilizar de manera conjunta con el operador de cruce. Primeramente se seleccionan dos individuos de la población para realizar el cruce. Si el cruce tiene éxito entonces uno de los descendientes, o ambos, se muta con cierta probabilidad Pm. Se imita de esta manera el comportamiento que se da en la naturaleza, pues cuando se genera la descendencia siempre se produce algún tipo de error, por lo general sin mayor trascendencia, en el paso de la carga genética de padres a hijos.

La probabilidad de mutación es muy baja, generalmente menor al 1 %. Esto se debe sobre todo a que los individuos suelen tener un ajuste menor después de mutados. Sin embargo se realizan mutaciones para garantizar que ningún punto del espacio de búsqueda tenga una probabilidad nula de ser examinado.

Tal y como se ha comentado, la mutación más usual es el reemplazo aleatorio. Este consiste en variar aleatoriamente un gen de un cromosoma. Si se trabaja con codificaciones binarias consistirá simplemente en negar un bit. También es posible realizar la mutación intercambiando los valores de dos alelos del cromosoma. Con otro tipo de codificaciones no binarias existen otras opciones:

- Incrementar o decrementar a un gen una pequeña cantidad generada aleatoriamente.
- Multiplicar un gen por un valor aleatorio próximo a 1.

Aunque no es lo más común, existen implementaciones de Algoritmos Genéticos en las que no todos los individuos tienen los cromosomas de la misma longitud. Esto implica que no todos ellos codifican el mismo conjunto de variables. En este caso existen mutaciones adicionales como puede ser añadir un nuevo gen o eliminar uno ya existente.

7. Evaluación

Para el correcto funcionamiento de un Algoritmo Genético se debe de poseer un método que indique si los individuos de la población representan o no buenas soluciones al problema planteado. Por lo tanto para cada tipo de problema que se desee resolver deberá derivarse un nuevo método, al igual que ocurrirá con la propia codificación de los individuos.

De esto se encarga la función de evaluación, que establece una medida numérica de la bondad de una solución. Esta medida recibe el nombre de ajuste. En la naturaleza el ajuste (o adecuación) de un individuo puede considerarse como la probabilidad de que ese individuo sobreviva hasta la edad de reproducción y se reproduzca. Esta probabilidad deberá estar ponderada con el número de descendientes. Evidentemente no es lo mismo una probabilidad de reproducción del 25 % en una población de un par de cientos de individuos que esa misma probabilidad en una población de varios millones.

En el mundo de los Algoritmos Genéticos se empleará esta medición para controlar la aplicación de los operadores genéticos. Es decir, permitirá controlar el número de selecciones, cruces, copias y mutaciones llevadas a cabo.

La aproximación más común consiste en crear explícitamente una medida de ajuste para cada individuo de la población. A cada uno de los individuos se les asigna un valor de ajuste escalar por medio de un procedimiento de evaluación bien definido. Tal y como se ha comentado, este procedimiento de evaluación será específico del dominio del problema en el que se aplica el Algoritmo Genético. También puede calcularse el ajuste mediante una manera 'co-evolutiva'. Por ejemplo, el ajuste de una estrategia de juego se determina aplicando esa estrategia contra la población entera (o en su defecto una muestra) de estrategias de oposición.

Se pueden diferenciar cuatro tipos de ajuste o fitness [Koza, 1992]:

- **Fitness Puro:** *r(i,t)*

  Es la medida de ajuste establecida en la terminología natural del propio problema. La ecuación 1 establece el cálculo del valor de bondad de un individuo i en un instante t (o generación).

$$r(i,t) = \sum_{j=1}^{N_c} |s(i,j) - c(i,j)| \tag{1}$$

Siendo :

$s(i,j)$ = valor deseado para el individuo i en el caso j

$c(i,j)$ = valor obtenido por el individuo i para el caso j

N_c = Número de casos

Por ejemplo, supóngase una población de hormigas que deben llenar la despensa de cara al invierno. La bondad de cada hormiga será el número de piezas de comida llevadas por ella hasta el hormiguero.

En los problemas de maximización, como sería el de las hormigas mencionado anteriormente, los individuos con un fitness puro elevado serán los más interesantes. Al contrario, en los problemas de minimización interesarán los individuos con un fitness puro reducido.

- **Fitness Estandarizado:** *s(i,t)*

  Para solucionar esta dualidad ante problemas de minimización o maximización se modifica el ajuste puro de acuerdo a la ecuación 2.

$$s(i,t) = \begin{cases} r(i,t) & \text{minimización} \\ r_{max} - r(i,t) & \text{maximización} \end{cases} \tag{2}$$

  En el caso de problemas de minimización se emplea directamente la medida de fitness puro. Si el problema es de maximización se resta de una cota superior r_{max} del error el fitness puro. Empleando esta métrica la bondad de un individuo será mayor cuanto más cercano esté a cero el valor del ajuste. Por lo tanto, dentro de la generación t, un individuo i siempre será mejor que uno j si se verifica que $s(i,t) < s(j,t)$.

- **Fitness Ajustado:** *a(i,t)*

  Se obtiene aplicando la transformación reflejada en la ecuación 3 al fitness estandarizado.

$$a(i,t) = \frac{1}{1 + s(i,t)} \tag{3}$$

  De esta manera, el fitness ajustado tomará siempre valores del intervalo [0..1]. Cuando más se aproxime el fitness ajustado de un individuo a 1 mayor será su bondad.

- **Fitness Normalizado:** *n(i,t)*

  Los diferentes tipos de fitness vistos hasta ahora hacen referencia únicamente a la bondad del individuo en cuestión. El fitness normalizado introduce un nuevo aspecto: indica la bondad de una solución con respecto al resto de soluciones representadas en la población. Considerando una población de tamaño N, se obtiene según la ecuación 4.

$$n(i,t) = \frac{a(i,t)}{\sum_{k=1}^{N} a(k,t)} \tag{4}$$

  Al igual que el fitness ajustado, siempre tomará valores del intervalo [0..1], con mejores individuos cuanto más próximo esté a la unidad. Pero a diferencia de antes, un valor cercano a 1 no sólo indica que ese individuo represente una buena solución al problema, sino que además es una solución destacadamente mejor que las proporcionadas por el resto de la población.

  La suma de los valores del fitness normalizado de todos los individuos de una población dará siempre 1.

  Este tipo de ajuste es empleado en la mayoría de los métodos de selección proporcionales al fitness.

Referencias

[Beasley et al., 1993] Beasley, D., Bull, D. R., and Martin, R. R. (1993). An overview of genetic algorithms: Part 1, fundamentals. *University Computing*, 15(2):58–69.

[Blickle and Thiele, 1995] Blickle, T. and Thiele, L. (1995). A comparison of selection schemes used in genetic algorithms. Technical Report 11, Computer Engineering and Comunnication Network Lab (TIK), Gloriastrasse 35, 8092 Zurich, Switzerland.

[Darwin, 1859] Darwin, C. (1859). *On the Origin of Species by Means of Natural Selection*. John Murray, London.

[Fogel et al., 1966] Fogel, L. J., Owens, A. J., and Walsh, M. J. (1966). *Artificial Intelligence through Simulated Evolution*. John Wiley & Sons, New York.

[Goldberg, 1989] Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[Holland, 1975] Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor. Republished by the MIT press, 1992.

[Jong, 1975] Jong, K. A. D. (1975). *An analysis of the behavior of a class of genetic adaptive systems*. PhD thesis, University of Michigan.

[Koza, 1992] Koza, J. R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection (Complex Adaptive Systems)*. The MIT Press.

[Rechenberg, 1973] Rechenberg, I. (1973). *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog, Stuttgart.

[Tomassini, 1995] Tomassini, M. (1995). A survey of genetic algorithms. *Annual Reviews of Computational Physics*, III:87–118.