Introducción

Los Algoritmos Genéticos (llamados en un principio como Planes Reproductivos) son una rama de la Inteligencia Artificial que resuelve problemas de búsqueda y optimización. Para esto aplica conceptos relacionado con la Teoría de la Evolución de Darwin para desarrollar programas:

- Imitar procesos adaptativos de los sistemas naturales.
- Recrear modelos que usen mecanismos de los sistemas naturales.
- Mecanismos de selección de acuerdo de la supervivencia del gen más apto al cambio.

El objetivo principal de los algoritmos genéticos es que las computadoras "aprendan" por sí solas, osea usar un algoritmo que se auto re-estructure conforme va corriendo el programa. Para esto se crea una posible solución y se la va haciendo converger a valores más óptimos, aplicando una serie de operadores similares a los de la evolución biológica natural. Esto proceso consiste en un rutina de software que toma una población y devuelve una nueva población reformada por estos operadores, reemplazando a la antigua población. Estos cambios que producen los operadores, si están correctamente aplicados, a la larga van a lograr que converjan hacia valores óptimos. Los operadores intentan imitar el proceso de reproducción sexual de los humanos.

Los algoritmos genéticos surgen como respuesta a la cuestión de cómo se puede imitar a la naturaleza, que logra seres cada vez más perfectos y mejores adaptados, la primer respuesta que obtiene es que esto es gracias a las interacciones que logra con otros individuos y también las que logra con el medio. Estos otros individuos y el medio en conjunto componente lo que se denomina como Ambiente de un individuo. Se dice que el ambiente ejerce una presión selectiva que hace que el individuo sufra cambios "en una cierta dirección".

Los AG están dirigidos por probabilidades, ya que es la mejor manera que tenemos de representar el comportamiento de la naturaleza -por ahora- , ya que en muchos aspectos se nos hace imposible de comprender.

Se definen conceptos paralelamente entre la realidad y la representación computacional:
Cromosoma: es la representación de un individuo u objeto real, las características están definidas por sus parámetros, a los que llamaremos genes.
Gen: representan a las variables reales, son la unidad básica de información de los AG, todas las operaciones que hagamos van a ser a nivel de genes. Una serie de genes componen un cromosoma (individuo).

Marco de Desarrollo

Los AG se engloban dentro de una ciencia que se conoce como Computación Evolutiva. Esta ciencia estudia las técnicas heurísticas aplicando fundamentos de evolución humana. Tiene varias ramas que se diferencian en cómo aplican estos fundamentos, ya que estos en sí son transversales a todas.

Por lo pronto, se puede definir algunos fundamentos:

• La evolución ocurre a nivel de genes y no de invidividuos. Cada individuo es representado por un conjunto de genes llamando cromosoma.
• La selección natural es el mecanismo por el cual los seres mejores adaptados tienen más probabilidad de reproducirse.
• El proceso evolutivo se da en la etapa de reproducción.

La Computación Evolutiva se divide en 5 grandes ramas:

- ***Los Algoritmos Genéticos***
- ***Programación Genética:*** se centra en estudios cuya solución es un programa. Los programas "compiten" entre sí para ver cual encuentra la mejor solución.
- ***Programación Evolutivas:*** es otro enfoque de los AG en el que se intenta conseguir que los operadores genéticos imiten los mejor posible a la naturaleza más que centrarse en la relación padre-hijo con su descendencia. El cruce pierde importancia y la gana la mutación.
- ***Estrategias Evolutivas:*** estudian problemas de optimización a nivel de genotipo y fenotipo.
- ***Sistemas Clasificadores:*** estudian problemas en los que la solución corresponde a una población y no a un individuo.

Ventajas de los AG:

- Operan con varias soluciones a la vez.
- Se ve menos afectados por los máximos locales que los métodos tradicionales.
- Fácil de ejecutar. En general, no requiere una gran despliegue de tecnología.
- Usar operadores probabilísticos.
- Resuelven problemas de varias áreas.
- Son, en gran medida, reutilizables en cuanto a la programación.
- Si no existe una técnica especializada de resolución, estos brindan soluciones de calidad y en tiempos aceptables.
- Se pueden hibridar con técnicas tradicionales.

Desventajas de los AG:

- Pueden tardar mucho en converger a buenas soluciones o no hacerlo nunca.
- Pueden converger prematuramente dándonos resultados malos o máximos locales.
- En general, si existe una técnica especializada para resolver un problema, el AG es menos efectivo y brinda peores resultados.

Comparación de los AG con otros métodos de optimización

Método de Optimización	Descripción	Ventajas de AG	Desventajas de AG
Algoritmos Matemáticos		Se pueden aplicar en funciones representativos no continuas. Pueden trabajar con todo tipo de funciones. Puede trabajar con funciones dinámicos.	Si puede ser resuelto por un algoritmo tradicional, casi seguro que brinda mejores resultados que un AG.
Métodos Enumerativos	Enumeran todas las soluciones y las analizan, por ejemplo, el problema de la mochila.	Pueden ser implementado incluso en grandes espacios de solución.	Reducen drásticamente el número de datos a utilizar. Si es posible usar un ME en un tiempo lógico, conviene hacerlo, ya que nos brinda un montón de soluciones reales y 100% concretas.
Sistemas Expertos	Encuentran soluciones condicionales. Esta condiciones deben ser provistas por especialistas. El conocimiento no estructurado se hacía imposible de manejar. Requiere que los conocimientos estén disponibles estructurados y que sean estáticos.	No se ven tan afectados por el cambio de conocimiento. Solo necesita la presencia de especialistas para hacer cambiar que no afecten los parámetros. Pueden trabajar con información inexacta y/o escasa.	Ideales para sistemas en entornos reducidos y con condiciones de ejecución acotadas.
Redes Neuronales	Simulan computacionalmente el comportamiento del derecho humano. Usa el concepto de capas, estas se transfieren información y la modifican a lo largo de la red.	La red puede ser entrenada para diversos usos, entre ellos, la optimización de funciones. En este aspecto compite con los AG, aunque como son de diferentes ramas no hay mucho material con respecto a la combinación.	

Hay una serie de condiciones que debe cumplir un problema para que le pueda ser aplicado un AG:

- Espacio de búsqueda finito. En caso contrario, hay que discretizar los espacios continuos.
- Debemos poder definir una Función Fitness que nos indique que tan buenas es esa solución.
- Las soluciones deben poder codificarse relativamente fácil.

Para hacer un AG primero debemos definir su arquitectura, sus componentes principales son:

- Tamaño de la población.
- Cantidad de generaciones y Condición de Terminación.
- Operadores utilizados (Selección con o sin Elitismo, Crossover, Mutación, entre otros) y sus parámetros.
- Función objetivo.
- Función Fitness.
- Codificación de los Cromosomas.

Tamaño de la población

Debe ser suficiente para que haya diversidad, las poblaciones pequeñas corren el riesgo de no cubrir adecuadamente el espacio de búsqueda. En contra parte, las poblaciones grandes llevan un gran coste computacional.

Si lo cromosomas son de longitud variable, el tamaño de la población debería variar conforme a este, para hacer esto se usan operadores especializados en cromosomas con longitud variable. Si no, se mantiene constante, que lo que por lo general se suele usar.

El tamaño óptimo N se define mediante la siguiente función: $N = e^{L}$, donde L es la longitud de los cromosomas.
Si se sigue esta regla al pie de la letra, este AG es imposible de implementar para longitudes mas o menos grandes. Por eso se comprueba también, pero por regla empírica, que para la práctica, en relación
costo-beneficio, lo ideal es usar entre 1 y 21 cromosomas como tamaño para la población.

Cantidad de generaciones y Condición de Terminación

Se define una Condición de Terminación que nos dice cuando "saldríamos satisfechos" con una solución, osea cuanto requerimos que sea buena una solución para terminar la ejecución de algoritmo. Cuando se produce esto, se dice que el algoritmo converge a un cierto valor. Por ejemplo, que la mitad de los cromosomas tengan el mismo valor o condiciones similares.

En la práctica, esta condición de terminación muchas veces produciría una ejecución infinita, es por eso que también se debe definir una cantidad de generaciones máximas que el algoritmos puede producir, osea cuantas iteraciones le permitimos como máximo realizar al algoritmo.

Función Objetivo

Es la función a optimizar, es deseable que sea "regular", significa que no tenga saltos abruptos, que para valores de x cercanos al punto a $F(x) \approx F(a)$

Si esta función tiene muchos óptimos locales, se nos complica la resolución del problema y vamos a tener que aplicar operadores especializados para evitar que el algoritmo converja a soluciones locales y no al óptimo máximo.

Si la función objetivo es muy compleja, se pueden usar aproximaciones con otras funciones más simples, siempre que esto resulte en un tiempo de ejecución más rápido y brinde resultados de calidad aceptable.

Se puede controlar la velocidad de convergencia reduciendo o expandiendo el rango de la función objetivo.

A veces esta función puede tener valores no validos que deben ser controlados en el AG, reduciendo así el espacio solución. Aquí se muestran 3 soluciones posibles a este punto:

- ***Absolutista:*** Se siguen ejecutando cruces y mutaciones hasta que la solución sea válida o se asigna una función Fitness igual a cero, para que esta solución no sea tenida en cuenta.
- ***Reconstrucción:*** A aquellos cromosomas que violan las restricciones, se los reconstruye con un operador especial.
- ***Penalización:*** Se asignan valores de penalización al violar cada regla posible. Luego se recuentan para ver cuál tiene la penalización más baja.

Función Fitness

Es una función que toma cierto valor en función de un cromosoma dependiendo de cuan bueno es como solución. La función matemática en sí depende de cada problema. Esta Función Fitness va a moldear los operadores y el proceder del AG para que tome cromosomas con buen Fitness, o lo que es equivalente, que sean buenas soluciones.

Codificación de los Cromosomas

Los cromosomas se representan como cadena de caracteres. Estos caracteres pertenecen a un cierto alfabeto. El alfabeto más utilizado es el binario, aunque también se pueden usar números reales, por ejemplo, que nos da una mayor dificultad a la hora de trabajarlos pero también mayor diversidad a la hora de aplicar los operadores. El alfabeto no tiene por qué ser numérico, pueden se letras o símbolos en general.

Para cada variable real, se utiliza una cierta cantidad de genes que representa una subcadena del cromosoma en sí. Es así que luego se puede decodificar estos genes para obtener el nuevo valor real.

Hay dos maneras de clasificar una codificación, la primera es si dependen o no dependen del orden en que se acomoden las variables (se clasifican en Basadas en el orden y No Basadas en el Orden), la segunda si el resultado del AG representa un solución en sí o necesita de una conversión no trivial, por ejemplo cuando el resultado del AG es un árbol o un grafo (se clasifican en Directa e Indirectas).

Operadores

Selección

Se encarga de seleccionar los cromosomas que van a ser utilizados por los otros operadores de acuerdo a un cierto parámetro, que suele ser el Fitness. Incluye en su proceso el calculo de la Función Fitness en sí. Normalmente involucra el azar, lo que en la implementación involucra variables del tipo Random.

Hay varias maneras de implementar este operadores, algunas de ellas son:

Basada en el Rango: Se mantiene un X% de la población y el otro (100 – X)% se reemplaza por otros más aptos. Este valor X es un parámetro fijo que se define de acuerdo a la velocidad de convergencia deseada.

Rueda de Ruleta: Se crea un pool genético en donde cada cromosoma ocupa un espacio proporcional a su Función Fitness. Luego se toman cromosomas al azar de este pool.

Selección de Torneo: Se escogen X cantidad de cromosomas al azar y nos quedamos con el de mayor Fitness (con esto nos referimos a que el cromosoma peor reemplaza al cromosoma peor). Si X es demasiado grande corremos el riesgo de dejar buenos cromosomas afuera rápidamente, y si es demasiado pequeño, de darle mucho tiempo de vida a malos cromosomas.

La selección puede implementar otro operador llamado Elitismo, que consiste en copiar directamente los mejores N elementos de la población vieja en la población inicial.

Crossover

Es el principal operador genético. Recombina genes de los cromosomas que fueron seleccionados en base a un parámetro llamado probabilidad de Crossover. Debe tener cuidado que la población no converja rápido ,ya que puede hacerlo en óptimos locales, o que tarde demasiado en converge (o no lo haga nunca). Para esto hay tomar un buen parámetro de probabilidad de Crossover.

Se decide si hacer el crossover o no viendo si una cierta variable random toma menor valor a la probabilidad de Crossover. Si da como resultado un “true”, los genes se parten en bloques y se combinan entre sí, esto se hace con buenos bloques, dando como resultado que los buenos bloques se perpetúen en el tiempo produciendo buenos cromosomas (Teorema de los Esquemas). Es proceso de Crossover representa a la parte más importante de la reproducción sexual en sí en donde se toma parte de los genes paternos y parte de los maternos.

Hay varias maneras de implementar un Crossover, algunas de ellas son:

N-puntos: Dos cromosomas se cortan en un punto "N" entre 0 y la longitud de los cromosomas (elegido de azar) dando como resultado 2 bloques en cada cromosoma. Luego se intercambian el contenido de los bloques generados.

Uniforme: Se sigue un patrón llamado máscara de cruce de longitud igual a la longitud de los cromosomas. Se generada en binario y al azar. En una posición X, entre 0 y la longitud de los cromosomas, si el contenido de la máscara de cruce es igual a 1 se intercambian los cromosomas en esa posición, si es un 0, no se lo hace.

Especializados: Ciertos problemas tienen restricciones que no nos permiten usar Crossover generales, por ejemplo, en el problema del viajante, no se pueden repetir la ciudad en un mismo recorrido, por lo que deben tomarse ciertos recaudos y hacer un Crossover especializados distinto al de otros problemas.

Mutación

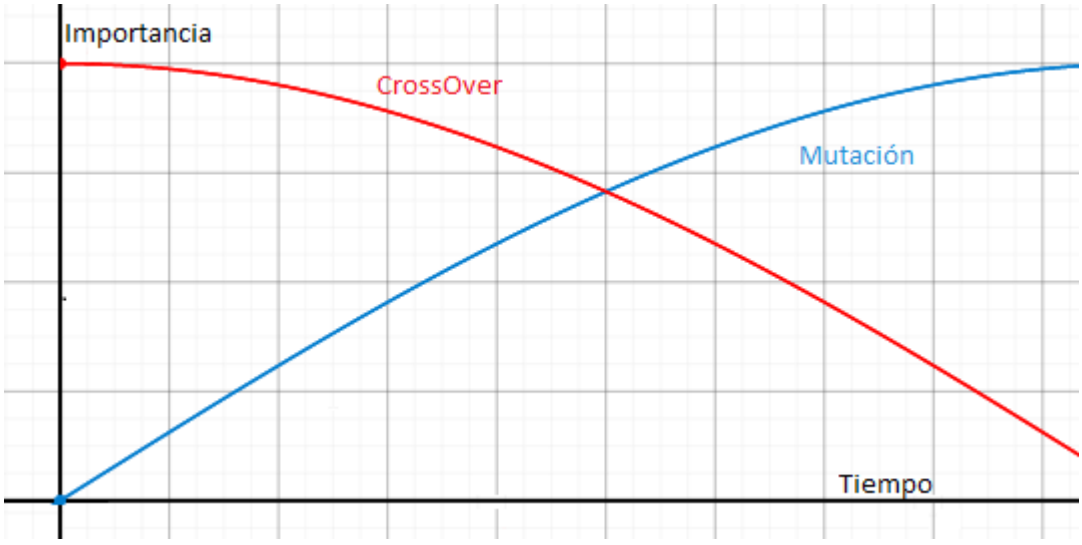
Esta operación muta los genes de los cromosomas de acuerdo a la probabilidad de mutación, es un suceso poco común y tiene un parámetro llamado probabilidad de Mutación que debe ser bajo. Contribuye a la diversidad de la especie (espacio solución), aunque si la probabilidad de Mutación no es pequeña se reduce la búsqueda a un búsqueda al azar, que claramente no es lo que estamos buscando. Evita lo convergencia prematura y ayuda a salir de los óptimos locales.

Nombre	Uso	Modo de Aplicación
Boundary Mutation	Codificaciones reales, enteras o finitas.	Elige una posición del cromosoma al azar y se reemplaza el contenido por el valor extremo del intervalo. Sirve para analizar bien los bordes de las funciones Objetivos.
Step Mutation	Codificaciones reales, enteras o finitas.	Mismo proceso que el anterior pero se reemplaza por el promedio entre el valor actual y el del borde.
Non Uniform Mutation	Codificaciones reales, enteras o finitas.	Mismo proceso que algunos de los anteriores, pero la probabilidad de Mutación se va reduciendo a medida que pasan las generaciones. Este valor va a 1 hacia 0.|
Non Uniform Mutation Modificada	Codificaciones reales, enteras o finitas.	Mismo proceso que al anterior, pero el valor de la probabilidad de Mutación aumenta de 0 a 1,|
Multi Non Uniform Mutation	Codificaciones reales, enteras o finitas.	Mismo proceso que Non Uniform Mutation pero afecta a todas las coordenadas del cromosoma y no una al azar.

Binary Mutation	Binario.	Se analiza gen por gen de cada cromosomas y se ve si un cierto Random es menor a la probabilidad de Mutación, si es así, se cambia el valor de ese gen en particular. Este método se llama mutación probabilística y la cantidad de mutaciones depende del azar. También puede decir que si la probabilidad de Mutación es de un X% que representa una cantidad Y de cromosomas del espacio solución, tomamos Y cantidad de genes al azar y los mutamos. Este otro método se llama mutación Determinística y se asegura que en cada iteración haya una cantidad Y de genes mutados siempre. Este último método tiene a ser más estable y seguro.

Relación Crossover-Mutación

El Crossover pierde utilidad a lo largo del tiempo porque los Cromosomas tienden a ser más parecidos o iguales. Con la mutación nos pasa lo contrario, va ganando importancia cuando la población va convergiendo. Se puede ir aumentado la probabilidad de mutación a medida que avanza hacia la convergencia. Aunque si la población se estanca, se convierte el proceso en una búsqueda aleatoria.



Operadores para Cromosomas con longitud variable

Deben implementar los operadores añadir gen y eliminar gen. Los nuevos genes suelen duplicarse de uno existente y luego mutarse.

Operadores de Nicho

Mantienen la diversidad de la población. Los cromosomas sustituyen solo a cromosomas similares.

Operadores Especializados

Restringen las soluciones por no ser válidas para un problema específico.

Algoritmo Genético Simple: Usa una codificación binaria, aplica crossover por el método de la ruleta y mutación binaria. Puede o no aplicar elitismo.

Algoritmos Nitching: Cuando queremos optimizar una función que tiene varios óptimos locales, un Algoritmos Genético común no nos sirve. Necesitamos usar una arquitectura especial para conservar todos los nichos ecológicos. Hay dos técnicas -entre otras- que sirven para este propósito:

- *Restricción de Deriva Genética:* Enlentecen el proceso restringiendo los cromosomas que pueden competir entre sí. De esta manera trabajamos con varios óptimos locales, evitando rápida convergencia a uno de ellos. Finalmente el óptimo global sí termina copando la población.

- *Entornos:* Se definen entornos con laderas y cimas. Las cimas son los posibles óptimos, las laderas con los sectores del costado. Una cima no tiene un su entorno un valor mas alto que ella. Se toma un punto, si no es una cima, se escala un poco hacia esta y se repite el proceso iterativamente. De esta manera también mantenemos lo óptimos locales.