



Algoritmos genéticos para encontrar el máximo en funciones

Neil Otniel Moreno Rivera

Universidad de Guanajuato, no.morenorivera@ugto.mx

Resumen— El problema a desarrollar un programa, que mediante el uso de algoritmos genéticos encuentre el punto máximo de una función, mediante una población binaria.

Abstract-- The problem is to develop a program that, through the use of genetic algorithms, finds the maximum point of a function, through a binary population.

I. INTRODUCCIÓN

Primero, debemos conocer que es un algoritmo genético. 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 (1859). 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.

II. TEORÍA

En la naturaleza los individuos de una población compiten entre sí en la búsqueda de recursos tales como comida, agua y refugio. Incluso los miembros de una misma especie compiten a menudo en la búsqueda de un compañero. Aquellos individuos que tienen más éxito en sobrevivir y en atraer compañeros tienen mayor probabilidad de generar un gran número de descendientes. Por el contrario, individuos poco dotados producirán un menor número de descendientes. Esto significa que los genes de los individuos mejor adaptados se propagaran en sucesivas generaciones hacia un número de individuos creciente. La combinación de buenas características provenientes de diferentes ancestros puede a veces producir descendientes “superindividuos”, cuya adaptación es mucho mayor que la de cualquiera de sus ancestros. De esta manera, las especies evolucionan logrando unas características cada vez mejor adaptadas al entorno en el que viven.[1]

El uso del comportamiento natural como analogía en algoritmos genéticos. Trabajan con un grupo de personas, cada una de las cuales representa una solución viable para un problema específico. De acuerdo con la calidad de la solución propuesta, a cada persona se le asigna un valor o puntuación. En el mundo natural, esto es equivalente a la eficacia de un organismo para competir por un recurso determinado. La probabilidad de que un

individuo sea elegido para reproducirse cruzando su material genético con otro individuo elegido de la misma manera aumenta con el grado de adaptación del individuo al problema. Las nuevas personas nacidas de esta unión heredarán algunos rasgos de sus padres y serán descendientes de los primeros. La probabilidad de que un individuo sea elegido para la reproducción y, como resultado, que su material genético se transmita de generación en generación, disminuye con la disminución de la adaptación. Al hacer esto, se crea una nueva población de posibles soluciones, reemplazando a la anterior y confirmando la intrigante propiedad de que contiene una mayor proporción de rasgos positivos que la población anterior. Como resultado, los rasgos positivos se transmiten de generación en generación entre la población. Se exploran las regiones más prometedoras del espacio de búsqueda, favoreciendo el cruce de los individuos más adaptados. La población gravitará hacia la mejor solución al problema si el algoritmo genético ha sido bien diseñado. La fuerza de los algoritmos genéticos proviene de su solidez; pueden resolver una amplia gama de problemas de varios campos, incluso aquellos en los que otros enfoques tienen problemas. Aunque no se garantiza que el Algoritmo Genético encuentre la mejor solución, existe evidencia empírica de que encontrará soluciones que son al menos tan buenas como las alternativas en un tiempo razonable.

Ahora con esta información, podemos explicar como resolver la problemática, esta nos pide los siguientes requisitos:

$$F(x) = 1 - x^2$$

* Neil Otniel Moreno Rivera.

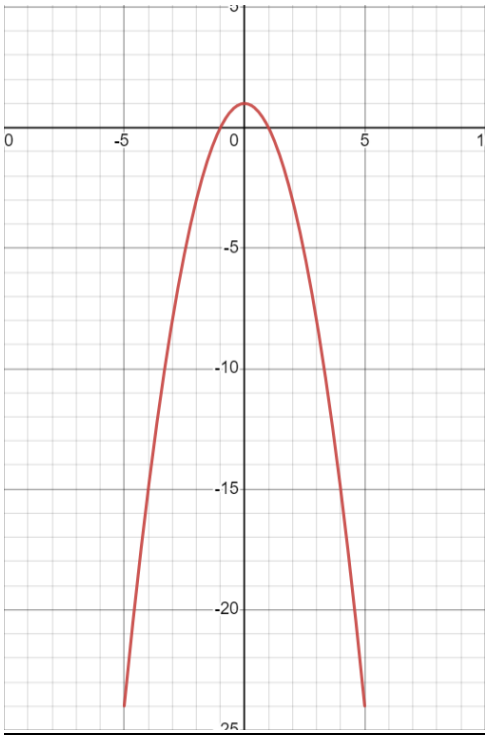


Fig 1. Grafica de la función dada por el problema.

Utilizando el método de selección de ruleta dentro de un rango de valores entre -5 y 5, se implementará un proceso de cruce con una probabilidad del 50%. No se realizarán mutaciones y el cambio generacional se llevará a cabo reemplazando a los padres por los hijos.

¿A que nos referimos con esto? Primero, el método de selección de ruleta es literalmente crear una ruleta, donde el tamaño de las secciones estará dado, de la siguiente forma, “Mientras mayor sea, más espacio le corresponde y por lo tanto, tendrá más posibilidades de ser elegido”, esto con todos los elementos de la población.

El rango, aunque sea obvio, es el espacio donde se trabajará. La probabilidad de cruce nos define el lugar donde se cortarán los padres para mezclarse y obtener valores para los hijos, esto se lleva acabo de la siguiente forma, lo siguiente es un ejemplo:

| Longitud del cromosoma= 8 bits | |
|--------------------------------|----------|
| Padre 1 | 01101111 |
| Padre 2 | 10101101 |
| Hijo 1 | 01101101 |
| Hijo 2 | 10101111 |

Con esto buscaremos una mejoría, aunque puede no ser siempre así, esto para nuestro tamaño de población que en este caso son 50 elementos, obteniendo 50 hijos de esos 50 padres y en este caso, fue para 100 generaciones, las generaciones las utilizamos para controlar las iteraciones que se realizan, esto es cuantas oleadas que hijos se obtendrán.

Con esto se realizo un programa de Python, comenzando en definir la función, después definir el espacio en el que se va a tratar, esto lo hacemos cambiando el enforque, puesto que el rango que nos proporciona el problema es de [-5,5], pero para nuestra resolución, borraremos los números del eje x, para colocar los nuestros, los nuestros serán una división dada por la

longitud del cromosoma, en este caso usamos 8 bits, por lo que el número máximo que podemos usar es el 256, por lo que se divide en 256 pedazos, dados en binario, para su posterior cruce. Desde ese punto, se colocan los elementos de la población, números aleatorios dentro del rango, en este caso 50 elementos, y llegando a este punto, usamos la ruleta, una ruleta dividida entre 50 pedazos, donde el grosor esta definido entre más grande sea.

III. RESULTADOS

Los resultados obtenidos, con los datos proporcionados por el problema fueron expresados en la siguiente figura, el resultado máximo o buscado fue el 1, como se muestra en la figura 1. Los datos que buscábamos era conocer la mejor solución encontrada, en este caso, ya convertida a decimal por facilidad es la mostrada en la figura 2, y al ser evaluada obtenemos el valor máximo, el cual debería ser uno, pero el resultado nunca es exacto, por lo que redondeado esta bien.

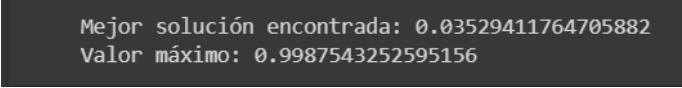


Fig 2. Imagen que muestra un aproximado de los datos de prueba perdidos.

IV. CONCLUSIONES

Para concluir con esto, puedo decir que los algoritmos genéticos, funcionan de una forma natural, por lo que no siempre es confiable o va a obtener el mejor resultado, puesto que esta abierto a la aleatoriedad por el método de ruleta, puesto que existen otros métodos más preciosos, como el de selección de máximos, en este caso, se uso el de ruleta, por lo que se usaron muchas generaciones, esto con el objetivo de asegurar un buen resultado.

El resultado obtenido por el programa, me parece correcto y hasta cierto punto fácil de entender.

ANOTACIONES

Este reporte fue realizado con lo que recordaba y lo re obtenido de fuentes que ya había visitado, si una información no se entiende, es porque no supe ordenar mis palabras.

REFERENCIAS

[1] Moujahid, Abdelmalik , et al. ALGORITMOS GENÉTICOS 1 Introducción.

BIBLIOGRAFÍA

1. Moujahid, Abdelmalik , et al. ALGORITMOS GENÉTICOS 1 Introducción.
2. Samir Roy and Udit Chakraborty, (2013) Soft Computing: Neuro-Fuzzy and Genetic Algorithms, Pearson.
3. P. Larranaga, M. Poza (1994). Structure learning of Bayesian network by genetic algorithms. E. Diday (ed.), New Approaches in Classification and Data Analysis, Springer-Verlag