INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

GENETIC ALGORITHMS

REPORTE PRACTICA III

"ALGORITMO GENÉTICO SIMPLE"

ALUMNO: SOLIS SANCHEZ JHOVANY

GRUPO: 3CM5

BOLETA: 2015630489

INTRODUCCIÓN

Los objetivo es implementar la técnica de selección proporcional o de ruleta, la cual fue propuesta por DeJong y ha sido el método más comúnmente usado desde los orígenes de los algoritmos genéticos.

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.

El algoritmo es simple, pero ineficiente ya que su complejidad es $O(n^2)$. Asimismo, presenta el problema de que el individuo menos apto puede ser seleccionado más de una vez. Sin embargo, buena parte de su popularidad se debe no sólo a su simplicidad, sino al hecho de que su implementación se incluye en el libro clásico sobre AGs de David Goldberg.

El algoritmo de la Ruleta es el siguiente:

• Calcular la suma de valores esperados **T**
• Repetir **N** veces (**N** es el tamaño de la población):

– Generar un número aleatorio **r** entre 0.0 y **T**
– Ciclar a través de los individuos de la población sumando los valores esperados hasta que la suma sea mayor o igual a **r**.
– El individuo que haga que esta suma exceda el limite es el seleccionado.

DESARROLLO

Con el fin de demostrar el funcionamiento del programa se utilizara una población de 8 individuos con 3 generaciones; posteriormente se utilizara una población de 32 individuos variando en 5, 10 y 15 generaciones que se obtendrán. Al iniciar la ejecución este generara una serie de individuos compuestos por 8 bits y obtendrá el su valor en decimal.

Individuos	x	x*x	Probabilidad	P. Acumulada
|1|1|0|1|0|1|0|0|	212	44944	0.2589	0.2589
|0|0|1|0|1|0|0|0|	40	1600	0.0092	0.2681
|0|1|0|1|0|1|0|1|	85	7225	0.0416	0.3097
|1|0|0|1|1|1|1|1|	159	25281	0.1456	0.4553
|0|1|0|1|0|1|0|0|	84	7056	0.0406	0.4959
|1|0|1|1|1|1|0|0|	188	35344	0.2036	0.6995
|0|1|0|1|0|1|0|1|	85	7225	0.0416	0.7411
|1|1|0|1|0|1|0|0|	212	44944	0.2589	1.0000

Illustration 1: Generación 0

Mostrara la el resultado de aplicarle la función de aptitud la cual para este algoritmo sera el cuadrado del valor decimal de cada individuo, mostrara la probabilidad o valor esperado a si como la probabilidad acumulada.

```
Suma : 173619
Promedio : 21702.3750
Probabilidad Maxima : 0.2589
Probabilidad Minima : 0.0092
```

Illustration 2: Generación 0 - datos

Realizara el calculo correspondiente a la suma total de los resultados obtenidos de la función aptitud, el promedio de estos, así como la probabilidad máxima y mínima de su generación.

```
Numero Aleatorio : 0.3700
Index : 3
Numero Aleatorio : 0.7200
Index : 6
Numero Aleatorio : 0.2800
Index : 2
Numero Aleatorio : 0.4300
Index : 3
Numero Aleatorio : 0.4400
Index : 3
Numero Aleatorio : 0.7300
Index : 6
Numero Aleatorio : 0.1200
Index : 0
Numero Aleatorio : 0.0600
Index : 0
```

Illustration 3: Números Aleatorios

Continuando con el algoritmo se generaran números coma flotantes aleatorios los cuales servirán para elegir un individuo; el individuo seleccionado sera aquel que en base a la probabilidad sobrepase el valor del numero aleatorio generado

```
Seleccionados          x
|1|0|0|1|1|1|1|1|      159
|0|1|0|1|0|1|0|1|      85
|0|1|0|1|0|1|0|1|      85
|1|0|0|1|1|1|1|1|      159
|1|0|0|1|1|1|1|1|      159
|0|1|0|1|0|1|0|1|      85
|1|1|0|1|0|1|0|0|      212
|1|1|0|1|0|1|0|0|      212
```

Illustration 4: Seleccionados

Los individuos seleccionados se cruzaran con el siguiente inmediato a el, la posición del bit a partir del cual se realizara la cruza sera generado nuevamente aleatoriamente de entre 0 y 8. En total se realizaran, con esta población de 8 individuos, 4 cruzas.

```
Cruzas Realizadas
Bit de cruza : 0
Bit de cruza : 3
Bit de cruza : 3
Bit de cruza : 2
```

Illustration 6: Cruzas

```
Cruza                  x
|0|1|0|1|0|1|0|1|      85
|1|0|0|1|1|1|1|1|      159
|0|1|0|1|1|1|1|1|      95
|1|0|0|1|0|1|0|1|      149
|1|0|0|1|0|1|0|1|      149
|0|1|0|1|1|1|1|1|      95
|1|1|0|1|0|1|0|0|      212
|1|1|0|1|0|1|0|0|      212
```

Illustration 5: Cruzas Realizadas

Después de realizar la cruza aplicara una mutación al 10% de la población lo cual hará que un bit en cualquier posición de algún individuo aleatorio cambie de valor, para esta población se cambiara el bit que esta en la 3er posición del segundo individuo quedando como resultado.

Mutacion	x
|0|1|0|1|0|1|0|1|	85
|1|0|1|1|1|1|1|1|	191
|0|1|0|1|1|1|1|1|	95
|1|0|0|1|0|1|0|1|	149
|1|0|0|1|0|1|0|1|	149
|0|1|0|1|1|1|1|1|	95
|1|1|0|1|0|1|0|0|	212
|1|1|0|1|0|1|0|0|	212

Illustration 7: Mutacion

El resultado para la primera generación sera el siguiente:

GENERACION : 1

Individuos	x	x*x	Probabilidad	P. Acumulada
|0|1|0|1|0|1|0|1|	85	7225	0.0369	0.0369
|1|0|1|1|1|1|1|1|	191	36481	0.1861	0.2229
|0|1|0|1|1|1|1|1|	95	9025	0.0460	0.2690
|1|0|0|1|0|1|0|1|	149	22201	0.1132	0.3822
|1|0|0|1|0|1|0|1|	149	22201	0.1132	0.4955
|0|1|0|1|1|1|1|1|	95	9025	0.0460	0.5415
|1|1|0|1|0|1|0|0|	212	44944	0.2293	0.7707
|1|1|0|1|0|1|0|0|	212	44944	0.2293	1.0000

Suma : 196046
Promedio : 24505.7500
Probabilidad Maxima : 0.2293
Probabilidad Minima : 0.0369

Illustration 8: Generacion 1

Este algoritmo se seguirá aplicando hasta cumplir las 3 generaciones, el programa cuenta la generación cero como una repetición de las que tiene que realizar.

En resultado final de las tres generaciones se observar la aparición de un candidato ideal (el segundo individuo) el cual se muestra a continuación:

```
GENERACION : 2

Individuos              x       x*x     Probabilidad    P. Acumulada
|1|0|0|1|0|1|0|0|       148     21904   0.0690          0.0690
|1|1|1|1|1|1|1|1|       255     65025   0.2048          0.2738
|1|0|1|1|1|1|1|0|       190     36100   0.1137          0.3876
|1|1|0|1|0|1|0|1|       213     45369   0.1429          0.5305
|1|0|0|1|0|1|0|1|       149     22201   0.0699          0.6004
|1|1|0|1|0|1|0|1|       213     45369   0.1429          0.7434
|1|0|1|1|1|1|1|0|       190     36100   0.1137          0.8571
|1|1|0|1|0|1|0|1|       213     45369   0.1429          1.0000
indD, int bits);
Suma : 317437
Promedio : 39679.6250
Probabilidad Maxima : 0.2048
Probabilidad Minima : 0.0690
```

Illustration 9: Resultado final

A continuación con este mismo algoritmo se utilizara una población de 32 individuos de 8 bits igualmente y se graficará los resultados de la probabilidad máxima y mínima de cada generación. Este proceso se realizara para 5, 10 y 15 generaciones.

5 Generaciones.

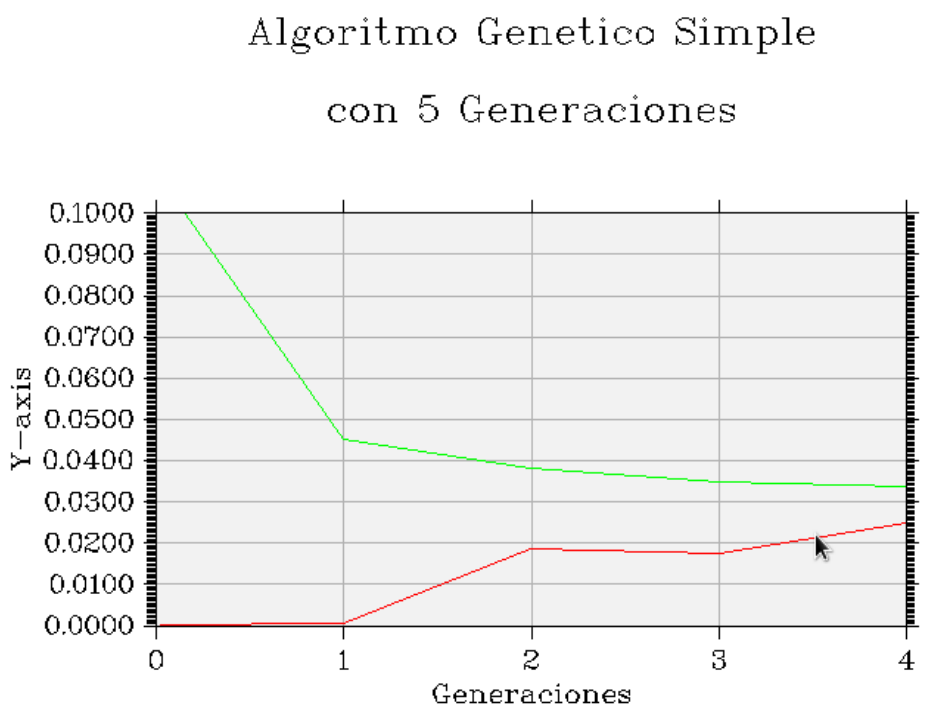

Illustration 10: 5 Generaciones

10 Generaciones.

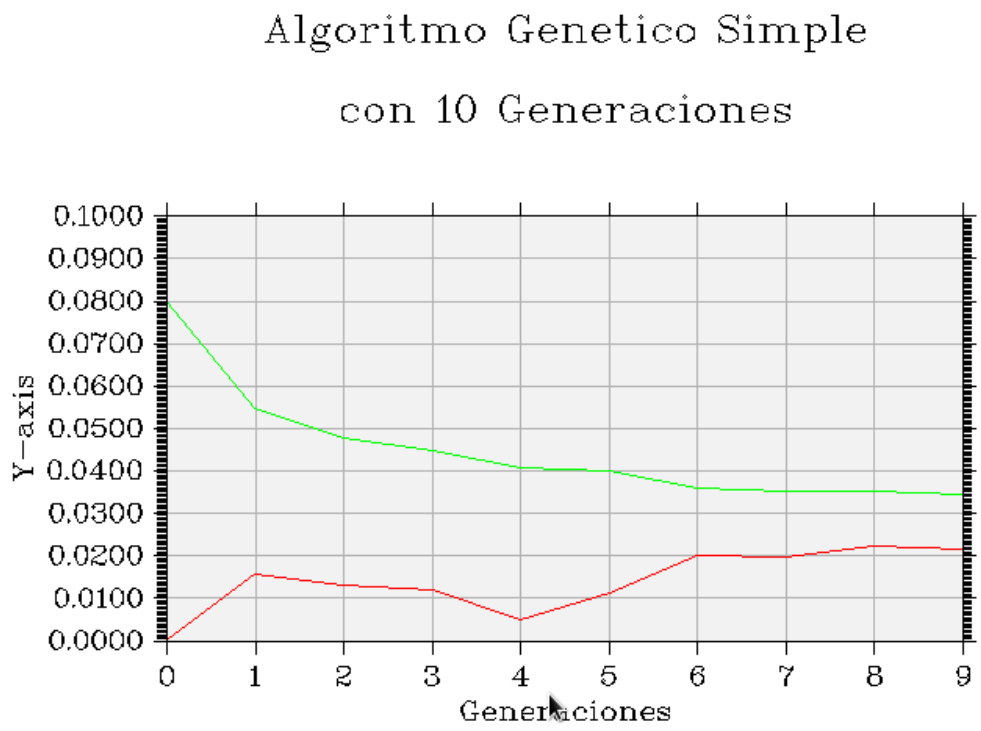

Illustration 11: 10 Generaciones

15 Generaciones.

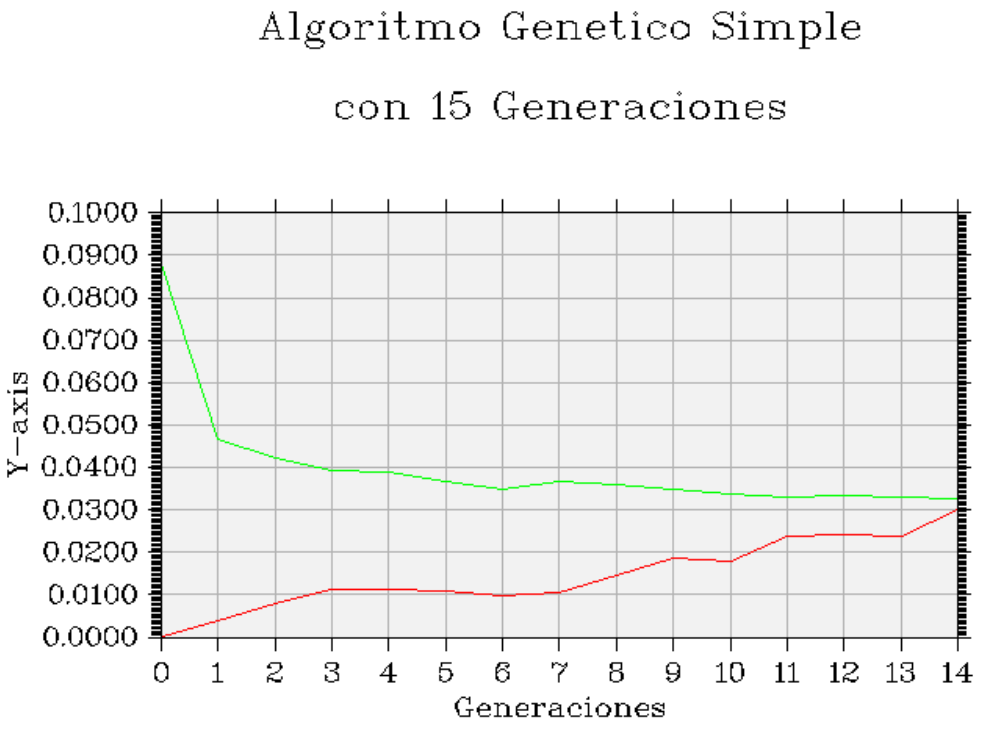

Illustration 12: 15 Generaciones

Como se puede observar tanto la gráfica el promedio máximo en color verde como el promedio mínimo en color rojo tienden a converger ambas gráficas conforme mas generaciones pasasen.

CONCLUSIÓN

Como se puede ver en algunos alguna gráficas se muestran algunos cambios en su trayectoria pero al final seguirán convergiendo mientras mas generaciones sean tomadas en cuenta, en algunos casos cuando la población aumentada al igual que las generación la maquina tardaba para realizar los cálculos y algunas veces mostraba algunas picos en las gráficas o cambiaba completamente la trayectoria de esta. A mi parecer la mutación provocaba esos cambios.