INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

GENETIC ALGORITHMS

REPORTE PRACTICA V

"SELECCIÓN POR JERARQUÍA"

ALUMNO: SOLIS SANCHEZ JHOVANY

GRUPO: 3CM5

BOLETA: 2015630489

INTRODUCCIÓN

Propuesta por Baker para evitar la convergencia prematura en las técnicas de selección proporcional. El objetivo de esta técnica es disminuir la presión de selección. En este caso, discutiremos el uso de jerarquías lineales, pero es posible también usar jerarquías no lineales, aunque la presión de selección sufre cambios más abruptos al usarse esta última.

Los individuos se clasifican con base en su aptitud, y se les selecciona con base en su rango (o jerarquía) y no con base en su aptitud. El uso de jerarquías hace que no se requiera escalar la aptitud, puesto que las diferencias entre las aptitudes absolutas se diluyen. Asimismo, las jerarquías previenen la convergencia prematura (de hecho, lo que hacen, es alentar la velocidad convergencia del algoritmo genético).

El algoritmo de las jerarquías lineales es el siguiente:

- Ordenar (o jerarquizar) la población con base en su aptitud, de 1 a N (donde 1 representa al menos apto).
- Elegir $Max(1 \leq Max \leq 2)$
- Calcular $Min = 2 - Max$
- El valor esperado de cada individuo será:

$$Valesp(i,t) = Min + (Max - Min)\frac{jerarquía(i,t)-1}{N-1}$$

Baker recomendó Max = 1.1
Usar selección proporcional aplicando los valores esperados obtenidos de la expresión anterior.

Complejidad: O(nlogn) + tiempo de selección.

Algunos puntos interesantes respecto a la aplicabilidad de esta técnica:

- Es útil cuando la función tiene ruido (p.ej., cuando hay una variable aleatoria).
- Diluye la presión de la selección, por lo que causa convergencia más lenta.
- Existen otros métodos de asignación de jerarquías además del lineal (p. ej.,exponencial).
- Puede alentar sobremanera la convergencia del algoritmo genético, por lo que su uso suele limitarse a situaciones en las que el AG convergería prematuramente en caso de no aplicarse.

DESARROLLO

Con el fin de demostrar el funcionamiento del programa se utilizara una población de 16 individuos variando la cantidad de generaciones de en 5, 30,50 y 100 generaciones que se obtendrán. Al iniciar la ejecución este generara una serie de individuos compuestos por 4 bits y obtendrá el su valor en decimal.

La aplicación muestra los individuos generados, su valor en binario y el resultado de la función finesses.
La cual se muestra continuación:

Individuos	x	F. finesses
|0|1|1|0|	6	0
|1|1|1|0|	14	3
|0|1|1|0|	6	0
|1|1|1|0|	14	3
|1|0|1|1|	11	5
|1|1|1|1|	15	3
|0|0|1|0|	2	1
|1|0|1|1|	11	5
|0|1|1|1|	7	0
|0|0|0|1|	1	1
|0|1|1|1|	7	0
|1|1|0|1|	13	3
|1|0|1|0|	10	3
|0|1|1|0|	6	0
|1|0|1|1|	11	5
|0|0|0|1|	1	1

Illustration 1: Individuos

$$f(x) = ABS\left|\frac{x-5}{2+Sen(x)}\right|$$

Illustration 2: Funcion Finesses

El mejor individuo sera aquel que logre maximizar la función finesse.
El algoritmo ordenara los individuos de mayor a menor para poder asignarle una posicion dentro de una jerarquía siendo el numero 1 el menos apto.
A todos los individuos pasaran por una función para obtener su valor esperado el cual estará acotado por un mínimo de 0.9 y un máximo de 1.1.
Y mostrara la probabilidad acumulada para cada individuo dado un total de 16.

Jerarquia	Individuo	x	ValEsp	P. Acumulada
1	|0|0|0|1|	1	0.9000	0.9000
2	|0|0|0|1|	1	0.9133	1.8133
3	|0|0|1|0|	2	0.9267	2.7400
4	|0|1|1|0|	6	0.9400	3.6800
5	|0|1|1|0|	6	0.9533	4.6333
6	|0|1|1|0|	6	0.9667	5.6000
7	|0|1|1|1|	7	0.9800	6.5800
8	|0|1|1|1|	7	0.9933	7.5733
9	|1|0|1|0|	10	1.0067	8.5800
10	|1|0|1|1|	11	1.0200	9.6000
11	|1|0|1|1|	11	1.0333	10.6333
12	|1|0|1|1|	11	1.0467	11.6800
13	|1|1|0|1|	13	1.0600	12.7400
14	|1|1|1|0|	14	1.0733	13.8133
15	|1|1|1|0|	14	1.0867	14.9000
16	|1|1|1|1|	15	1.1000	16.0000

Illustration 3: Jerarquias

Se generaran números aleatoriamente de 1 a 16 para poder seleccionar a los posibles individuos candidatos a ser solución estos se hará hasta completar los nuevos 16 individuos.

```
Numero Aleatorio : 12.0000
Index : 12
Numero Aleatorio : 6.0000
Index : 6
Numero Aleatorio : 8.0000
Index : 8
Numero Aleatorio : 1.0000
Index : 1
Numero Aleatorio : 8.0000
Index : 8
Numero Aleatorio : 4.0000
Index : 4
Numero Aleatorio : 5.0000
Index : 5
Numero Aleatorio : 5.0000
Index : 5
Numero Aleatorio : 8.0000
Index : 8
Numero Aleatorio : 3.0000
Index : 3
Numero Aleatorio : 13.0000
Index : 13
```

Illustration 5: Seleccion de individuos

Seleccionados	x
|1|1|0|1|	13
|0|1|1|1|	7
|1|0|1|0|	10
|0|0|0|1|	1
|1|0|1|0|	10
|0|1|1|0|	6
|0|1|1|0|	6
|0|1|1|0|	6
|1|0|1|0|	10
|0|1|1|0|	6
|1|1|1|0|	14
|0|1|1|0|	6
|0|1|1|0|	6
|0|1|1|0|	6
|1|0|1|1|	11
|1|0|1|1|	11

Illustration 4: Individuos Seleccionados

Realizara al igual que la selección por ruleta una cruza los individuos de la población así como una mutación en el 30% de la población. Esto se repetirá según el número de generaciones que tomemos en cuenta.

```
Cruzas Realizadas
Bit de cruza : 2
Bit de cruza : 3
Bit de cruza : 0
Bit de cruza : 3
Bit de cruza : 1
Bit de cruza : 0
Bit de cruza : 3
Bit de cruza : 2
```

Illustration 8: Cruza

Cruza	x
|1|1|1|1|	15
|0|1|0|1|	5
|1|0|1|1|	11
|0|0|0|0|	0
|0|1|1|0|	6
|1|0|1|0|	10
|0|1|1|0|	6
|0|1|1|0|	6
|1|1|1|0|	14
|0|0|1|0|	2
|0|1|1|0|	6
|1|1|1|0|	14
|0|1|1|0|	6
|0|1|1|0|	6
|1|0|1|1|	11
|1|0|1|1|	11

Illustration 6: Cruza de Individuos

Mutacion	x
|1|1|1|1|	15
|0|1|0|1|	5
|1|0|1|1|	11
|0|0|0|0|	0
|0|1|1|0|	6
|1|0|1|0|	10
|0|1|1|0|	6
|0|1|1|0|	6
|1|1|1|0|	14
|0|0|1|0|	2
|0|1|1|0|	6
|1|1|1|0|	14
|0|1|1|0|	6
|0|1|1|0|	6
|1|0|1|1|	11
|1|0|1|1|	11

Illustration 7: Mutacion

Entre mas generaciones pasen puede ocurrir que los individuos se vean afectados en gran medida por las mutaciones que se realizan en cada generación.

5 Generaciones.

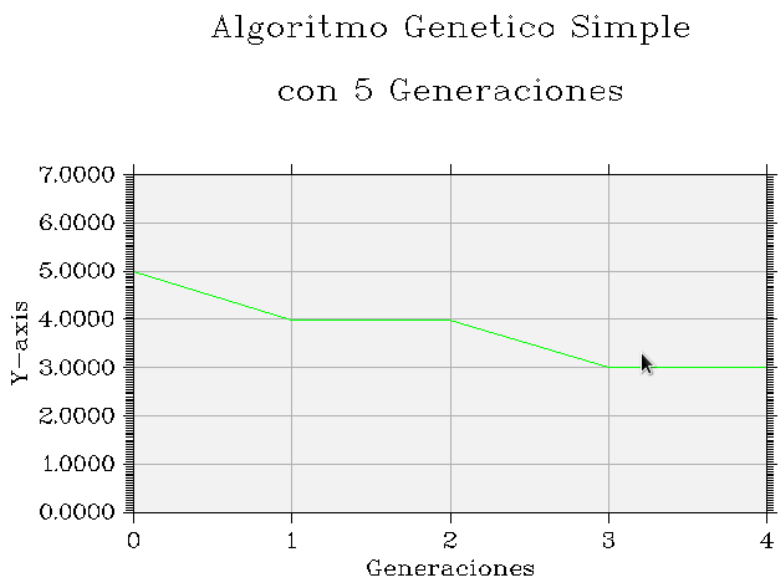

Illustration 9: 5 Generaciones

10 Generaciones.

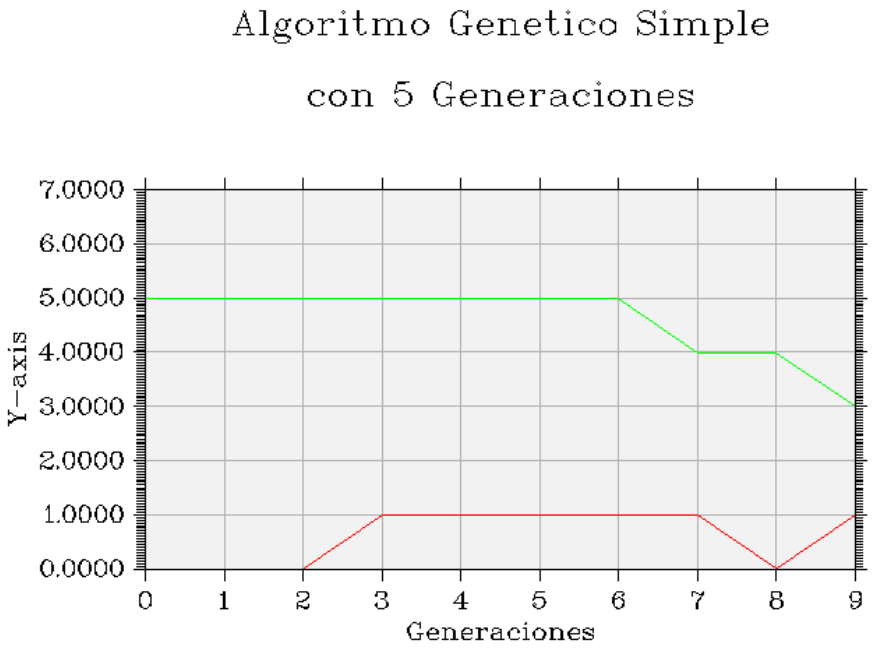

Illustration 10: 10 Generaciones

30 Generaciones.

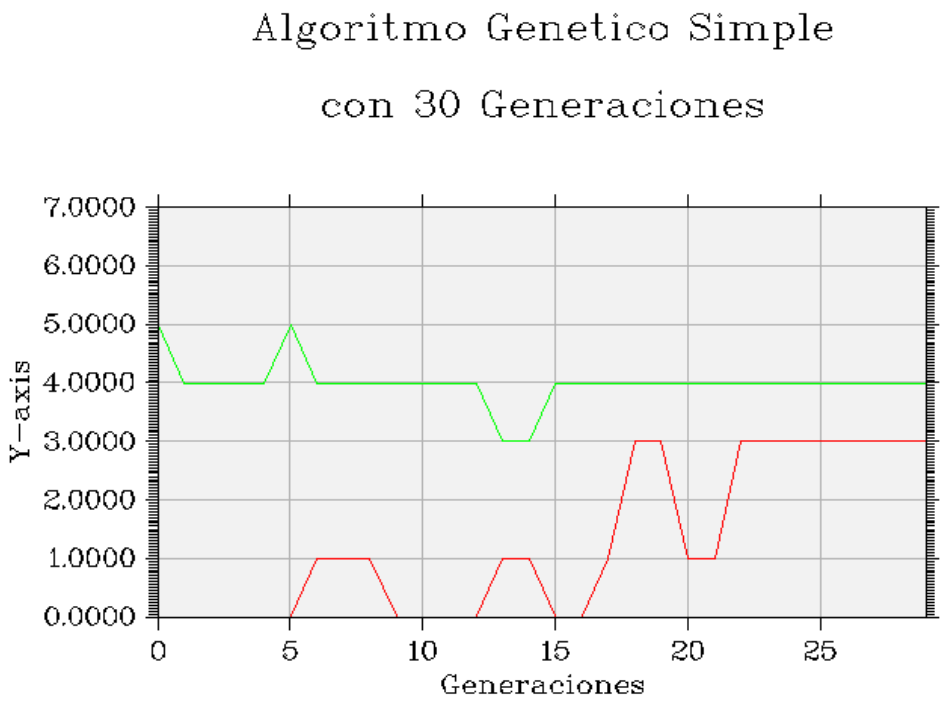

Illustration 11: 30 Generaciones

50 Generaciones

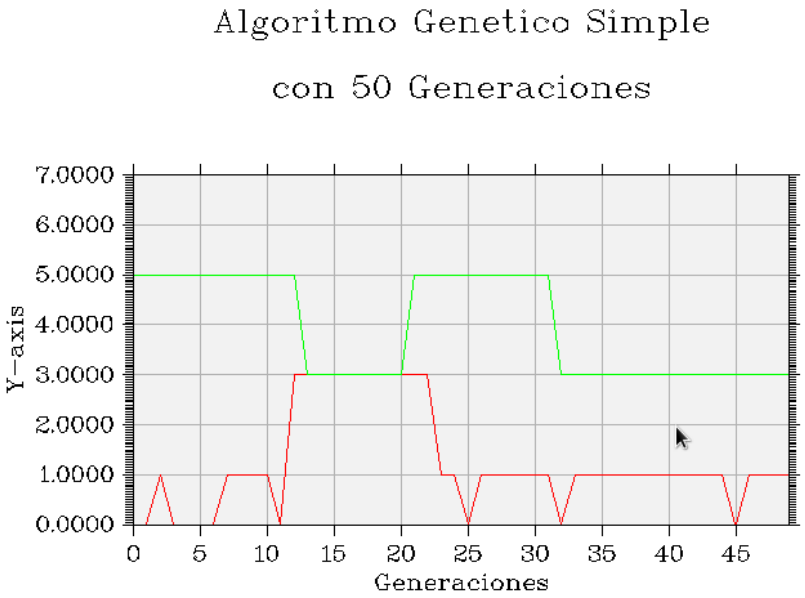

Illustration 12: 50 Generaciones

100 Generaciones

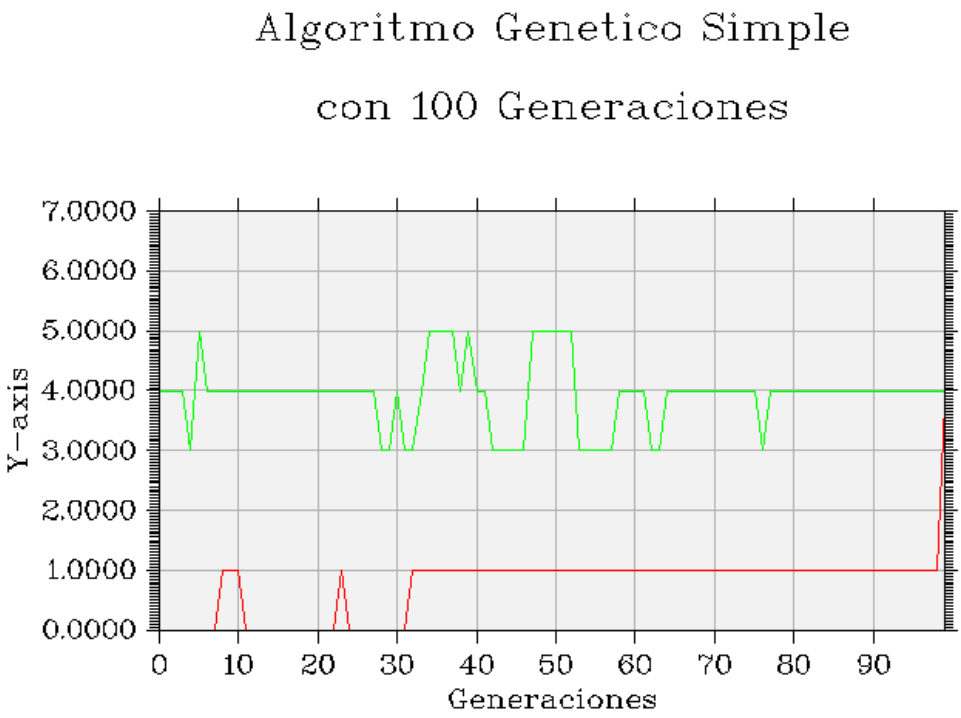

Illustration 13: 100 Generaciones

Como se puede observar tanto la gráfica el promedio máximo en color verde como el valor mínimo en color rojo de la función finesses estos tienden a converger ambas gráficas conforme mas generaciones pasasen.
Cuanto mas grande es la cantidad de generaciones a tomar en cuenta el algoritmo empieza a realizar cambios extraños en los individuos.

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. El algoritmo tendré que mejorar para resolver algunos conflictos de memoria que se provocan en alguna ocasiones.