



**INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO**



Algoritmos genéticos

“Selección por Ruleta”

Alumno:

Navarro Pérez Rafael 3CM5

Índex

Contenido

Introducción:	1
Desarrollo:	1
Resultados:.....	3
Conclusiones:	5

Introducción:

Selección por Ruleta

¿Qué es?

Esta técnica fue propuesta por De Jong [137], y ha sido el método más comúnmente usado desde los orígenes de los algoritmos genéticos. El algoritmo es simple, pero ineficiente (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 solo a su simplicidad, sino al hecho de que su implementación se incluye en el libro clásico sobre AGs de David Goldberg [105]. Un histograma muestra la acumulación o tendencia, la variabilidad o dispersión y la forma de la distribución.

El algoritmo de la Ruleta es el siguiente:

1. Calcular la suma de valores esperados T
2. Repetir N veces (N es el tamaño de la población):
 - Generar un numero 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 límite es el seleccionado.

Desarrollo:

Para el desarrollo de esta práctica se ocupó el IDE CodeBlocks, así como también se ocupó una librería grafica llamada ALLEGRO para poder realizar la práctica.

Para la implementación de algoritmo se ocuparon arreglos de bits para así poder hacer las operaciones necesarias, como lo son la selección, cruza y mutación.

En la práctica se crean aleatoriamente una población inicial de 5 bits cada individuo.

Posteriormente sigue la etapa de la selección en este caso se ocupó el algoritmo de La Ruleta que se explicó a un principio, se guarda en un nuevo arreglo de bits el cual se llama selección y es el que se ocupa para la siguiente etapa, la cruza, que da como resultado un nuevo arreglo de bit el cual por último se hace la mutación y se pasa a la siguiente generación hasta llegar a la generación deseada.

Se crearon las siguientes funciones:

```
1. void IniciarInd(bitset<BIT_IND> array[]); //da valor inicial a todo el arreglo
2.
3. void printIndividuals(bitset<BIT_IND> array[]); //imprime los individuos
4.
```

```

5. int getIndividualValue(bitset<BIT_IND> individual);//retorna el valor del individuo
6.
7. int getIndividualAptitude(bitset<BIT_IND> individual);//retorna la aptitud
8.
9. int getTotalAptitude(bitset<BIT_IND> array[]);// retorna la sumatoria de aptitudes
   del arreglo
10.
11. bitset<BIT_IND> rouletteSelection(bitset<BIT_IND> array[], int totalAptitude);//al
    goritmo de Seleccion
12.
13. bitset<BIT_IND> crossAlgorithm(bitset<BIT_IND> &p1, bitset<BIT_IND> &p2);//cruza d
    e individuos
14.
15. bitset<BIT_IND> mutationAlgorithm(bitset<BIT_IND> individual);//mutacion de algori
    tmos
16.
17. int getMinGenerationValue(bitset<BIT_IND> array[]);//retorna el minimo valor de la
    generacion
18.
19. int getMaxGenerationValue(bitset<BIT_IND> array[]);//retorna el maximo valor de la
    generacion
20.
21. int getGenerationAverage(bitset<BIT_IND> array[]);//Da el promedio de generacion

```

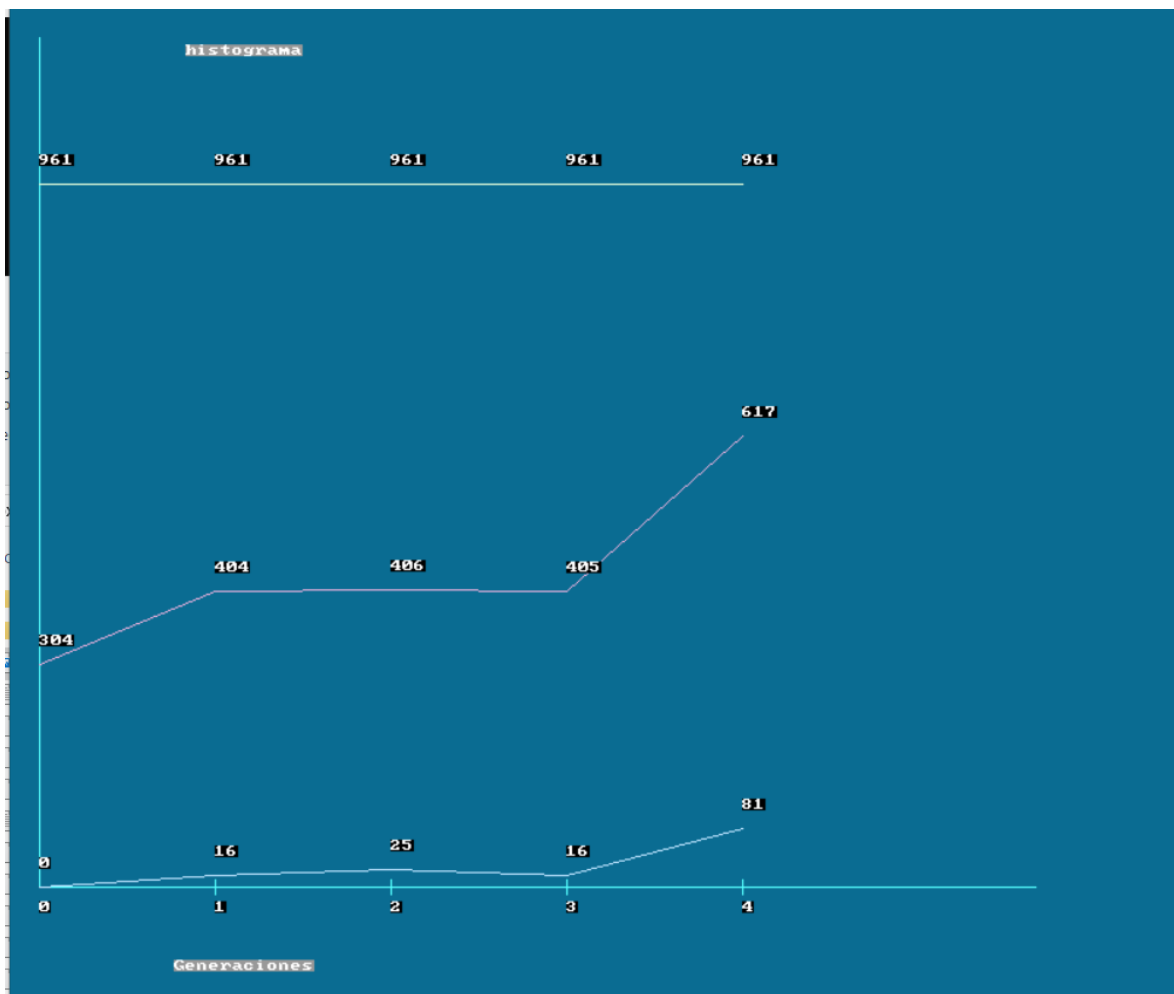
para el manejo de los arreglos de bits, y poder así visualizar el comportamiento de cada generación.

Donde en esta práctica lo que tiene mayor relevancia sería la función de rouletteSeleccction ya que es el objetivo de la misma.

Resultados:

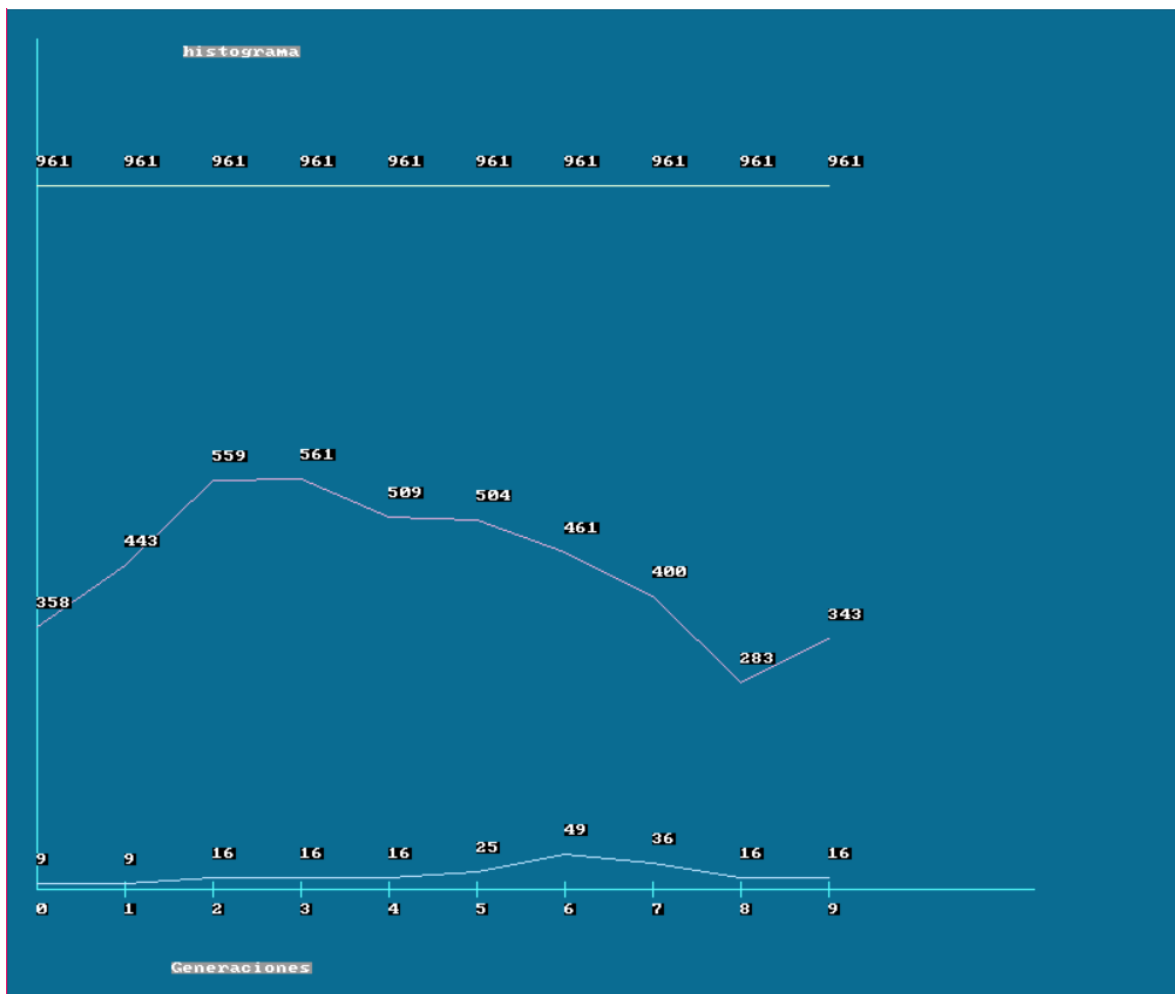
Como en la práctica se pedían cierta cantidad de generaciones al ejecutar el programa se podrá introducir la cantidad de generaciones que desees, para esta práctica los datos generados son 5, 10 y 15 generaciones.

Por lo cual empezaremos con la primera prueba que sería con un máximo de 5 generaciones.



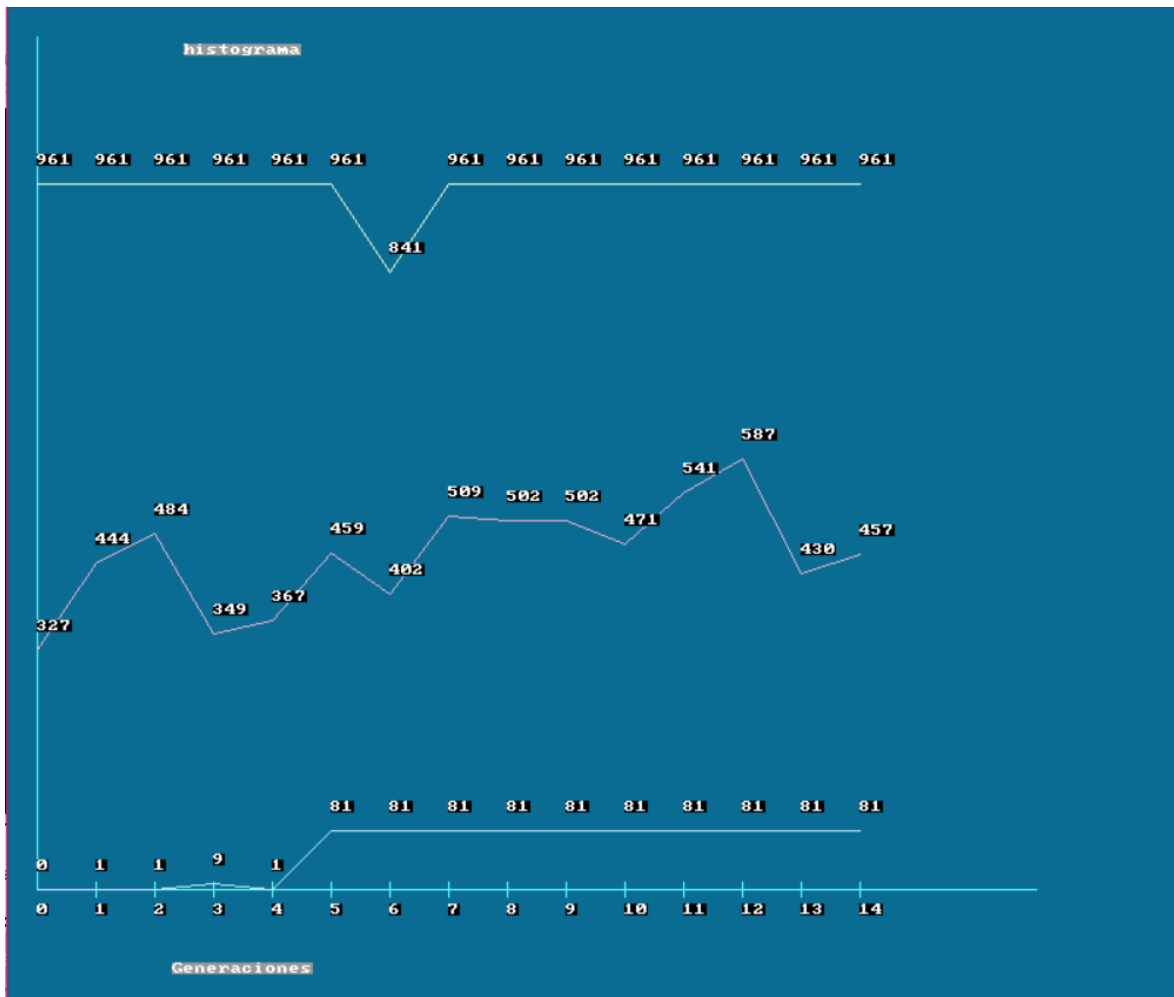
En el documento 5g.txt se puede observar el comportamiento de todos los individuos.

A continuación, con 10 generaciones:



Como se puede observar en este caso si conservo el individuo mas apto pero el individuo menos apto hubo ciertas variaciones.

Y por ultimo 15 generaciones:



Como se observa en la generación número se observa que el mejor individuo se elimino por la cruza con otro individuo o por la selección. Pero al empezar la siguiente generación con la cruza se recuperó y así poder mejorar e incluso el individuo menos apto fue mejorando poco a poco.

Conclusiones:

En esta práctica se pudo observar, que no por tener un individuo perfecto vaya a ser seleccionado como padre para poder ser heredado en las siguientes generaciones ya que se puede perder en la selección o cruza ya que se ven involucrados factores externos a cada individuo. Así también se puede observar con el paso de las generaciones que en efecto este algoritmo empieza a converger con forme vayan pasando más generaciones dando como resultado, individuos más aptos eliminando los que tienen una aptitud más bajas.