

**Instituto Politécnico Nacional**  
**Escuela Superior de Cómputo**



# **Practica 3:**

## **Selección por Ruleta**

**Genethic Algorithms**

**Álvarez González Oscar**

**Prof. Morales Guitaron Sandra Luz**

**Grupo: 3CM5**

# Índex

## Contenido

Introducción:

Desarrollo:

Resultados:

Conclusiones:

## Introducción:

Esta técnica fue propuesta por DeJong, 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.

El algoritmo de la Ruleta (de acuerdo a DeJong) es el siguiente:

1. Calcular la suma de valores esperados  $T$
2. 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 límite es el seleccionado.

Veamos ahora un ejemplo:

	aptitud	Ve
(1)	25	0.35
(2)	81	1.13
(3)	36	0.51
(4)	<u>144</u>	<u>2.01</u>
	$\sum = 286$	$\sum = 4.00$

$$\bar{f} = \frac{286}{4} = 71.5 \quad Ve_i = \frac{f_i}{\bar{f}}$$

$$T = \text{Suma de } Ve \\ r \in [0.0, T]$$

Generar  $r \in [0.0, 4.0]$

$$r = 1.3$$

(ind1) suma = 0.35 <  $r$

(ind2) suma = 1.48 >  $r$

Seleccionar a ind2

En este ejemplo,  $Ve$  se refiere al valor esperado (o número esperado de copias que se esperan obtener) de un individuo.

## Desarrollo:

Para el desarrollo de esta práctica se ocupó IDE Dev C++ el lenguaje usado en esta fue C y para la gráfica se utilizó IDE CodeBlocks ya que se utilizó la librería ALLEGRO para poder realizar la gráfica.

Para esta práctica se crearon las siguientes funciones:

```
void binario(int gen);
void inicializacion(int gen,unsigned char arrg[32][5]);
void seleccionRuleta(int gen, unsigned char arrg[32][5],float *Prob,int *dec,int *apt);
void cruza(int gen,unsigned char sel[32][5]);
void mutacion(int gen,unsigned char cruza[32][5]);
void grafica(float *Max, float *Min);
```

Ahora paso a explicar cada una de las funciones con una sola generación.

En la función binario solo se crea los individuos iniciales para que se valla realizando el algoritmo, esta función fue la misma que cree de la practica anterior, se hacen 32 arreglos de 5 bits cada uno, cada arreglo se creo al azar por lo que el valor máximo binario puede ser 31 y el mínimo 0.

En la función Inicialización lo único que hago es sacar el numero decimal del arreglo binario y con eso sacó la aptitud que en el caso de esta práctica fue  $f(x)=x^2$ , hago la sumatoria de la aptitud y sacó la probabilidad que en el

```
C:\Users\oscar\OneDrive\Escuela\Geneticos\Practicas\P3\C\P3
Generacion 1
Inicio
No || Pob. Ini || X || Aptitud X || Prob
0 || 11100 || 28 || 784 || 0.107
1 || 10100 || 20 || 400 || 0.054
2 || 11000 || 24 || 576 || 0.078
3 || 10000 || 16 || 256 || 0.035
4 || 00101 || 5 || 25 || 0.003
5 || 11001 || 25 || 625 || 0.085
6 || 01010 || 10 || 100 || 0.014
7 || 00010 || 2 || 4 || 0.001
8 || 01111 || 15 || 225 || 0.031
9 || 10101 || 21 || 441 || 0.060
10 || 00000 || 0 || 0 || 0.000
11 || 00100 || 4 || 16 || 0.002
12 || 01001 || 9 || 81 || 0.011
13 || 10100 || 20 || 400 || 0.054
14 || 01110 || 14 || 196 || 0.027
15 || 10011 || 19 || 361 || 0.049
16 || 01100 || 12 || 144 || 0.020
17 || 10001 || 17 || 289 || 0.039
18 || 01111 || 15 || 225 || 0.031
19 || 01110 || 14 || 196 || 0.027
20 || 10100 || 20 || 400 || 0.054
21 || 10111 || 23 || 529 || 0.072
22 || 00100 || 4 || 16 || 0.002
23 || 00010 || 2 || 4 || 0.001
24 || 00100 || 4 || 16 || 0.002
25 || 01001 || 9 || 81 || 0.011
26 || 00000 || 0 || 0 || 0.000
27 || 01000 || 8 || 64 || 0.009
28 || 01100 || 12 || 144 || 0.020
29 || 01110 || 14 || 196 || 0.027
30 || 10110 || 22 || 484 || 0.066
31 || 01001 || 9 || 81 || 0.011

Sumatoria Aptitud: 7359
Sumatoria Probabilidad: 1.00
```

En la función de seleccionRuleta se hacen las selecciones de los individuos que se seleccionaran para ello se genera un numero al azar “r” y se va sumando los individuos hasta que la suma de esos supere al numero “r”, en el ejeplo fue la selección 1 ya que fue corta y el individuo seleccionado fue el 2

```

1

r= 0.217

0 sum 0.000 +
0.107 = 0.107

1 sum 0.107 +
0.054 = 0.161

2 sum 0.161 +
0.078 = 0.239

2

11000

```

La función Cruza utiliza Cruza de un punto, selecciona al azar un punto del individuo y lo que sigue de ellos se intercambia con el siguiente individuo esta cruza se hace en pares de individuos.

```

01110Cruza
No || Cruza || P.Cruza || Descendencia || X || Aptitud X
0 || 01100 || 2 || 01000 || 8 || 64
1 || 11000 || 2 || 11100 || 28 || 784
2 || 10110 || 3 || 10110 || 22 || 484
3 || 10110 || 3 || 10110 || 22 || 484
4 || 11000 || 2 || 11100 || 28 || 784
5 || 11100 || 2 || 11000 || 24 || 576
6 || 11000 || 1 || 11100 || 28 || 784
7 || 01100 || 1 || 01000 || 8 || 64
8 || 11100 || 4 || 11100 || 28 || 784
9 || 01100 || 4 || 01100 || 12 || 144
10 || 11000 || 3 || 11001 || 25 || 625
11 || 00101 || 3 || 00100 || 4 || 16
12 || 11001 || 3 || 11001 || 25 || 625
13 || 10101 || 3 || 10101 || 21 || 441
14 || 01001 || 3 || 01000 || 8 || 64
15 || 01100 || 3 || 01101 || 13 || 169
16 || 11000 || 3 || 11001 || 25 || 625
17 || 11001 || 3 || 11000 || 24 || 576
18 || 10000 || 3 || 10011 || 19 || 361
19 || 10111 || 3 || 10100 || 20 || 400
20 || 01110 || 2 || 01110 || 14 || 196
21 || 10110 || 2 || 10110 || 22 || 484
22 || 01111 || 2 || 01001 || 9 || 81
23 || 01001 || 2 || 01111 || 15 || 225
24 || 10100 || 3 || 10100 || 20 || 400
25 || 10100 || 3 || 10100 || 20 || 400
26 || 10110 || 1 || 11001 || 25 || 625
27 || 11001 || 1 || 10110 || 22 || 484
28 || 11000 || 2 || 11011 || 27 || 729
29 || 10011 || 2 || 10000 || 16 || 256
30 || 01001 || 3 || 01010 || 10 || 100
31 || 01110 || 3 || 01101 || 13 || 169
Sumatoria Aptitud: 13003

```

La función mutación usa una mutación del 10% por lo que se pone al azar 3 unos de toda la descendencia.

Mutacion

No	Desc	Mutacion	X	Aptitud X	Prob
0	01000	01000	8	64	0.005
1	11100	11100	28	784	0.057
2	10110	10110	22	484	0.035
3	10110	10110	22	484	0.035
4	11100	11100	28	784	0.057
5	11000	11000	24	576	0.042
6	11100	11100	28	784	0.057
7	01000	01000	8	64	0.005
8	11100	11100	28	784	0.057
9	01100	01100	12	144	0.010
10	11001	11001	25	625	0.046
11	00100	00100	4	16	0.001
12	11001	11001	25	625	0.046
13	10101	10101	21	441	0.032
14	01000	01000	8	64	0.005
15	01101	01101	13	169	0.012
16	11001	11001	25	625	0.046
17	11000	11010	26	676	0.049
18	10011	10011	19	361	0.026
19	10100	10100	20	400	0.029
20	01110	01110	14	196	0.014
21	10110	10110	22	484	0.035
22	01001	01001	9	81	0.006
23	01111	01111	15	225	0.016
24	10100	10101	21	441	0.032
25	10100	10100	20	400	0.029
26	11001	11001	25	625	0.046
27	10110	10110	22	484	0.035
28	11011	11011	27	729	0.053
29	10000	10000	16	256	0.019
30	01010	11010	26	676	0.049
31	01101	01101	13	169	0.012

13720

Por último tenemos la función de grafica ahí se envían el max y el min con base en la probabilidad de cada generación para poder graficar.

```

maximo 0.057
minimo 0.001

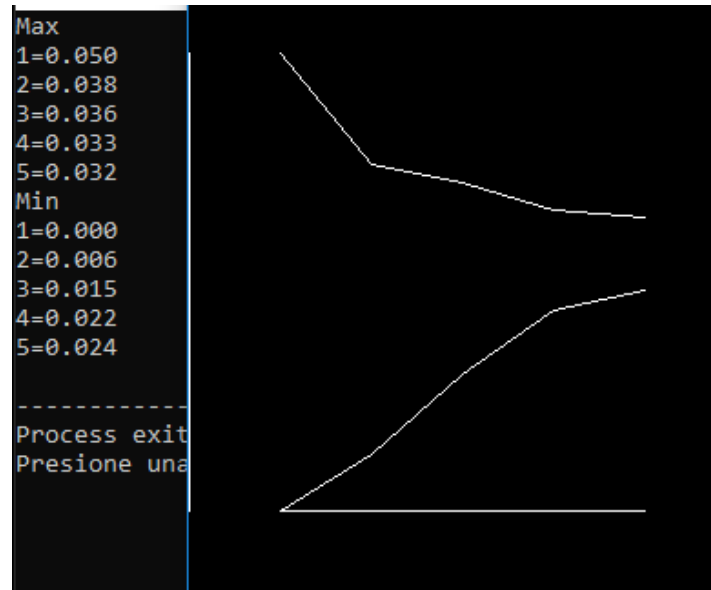
Max
1=0.057
Min
1=0.001

```

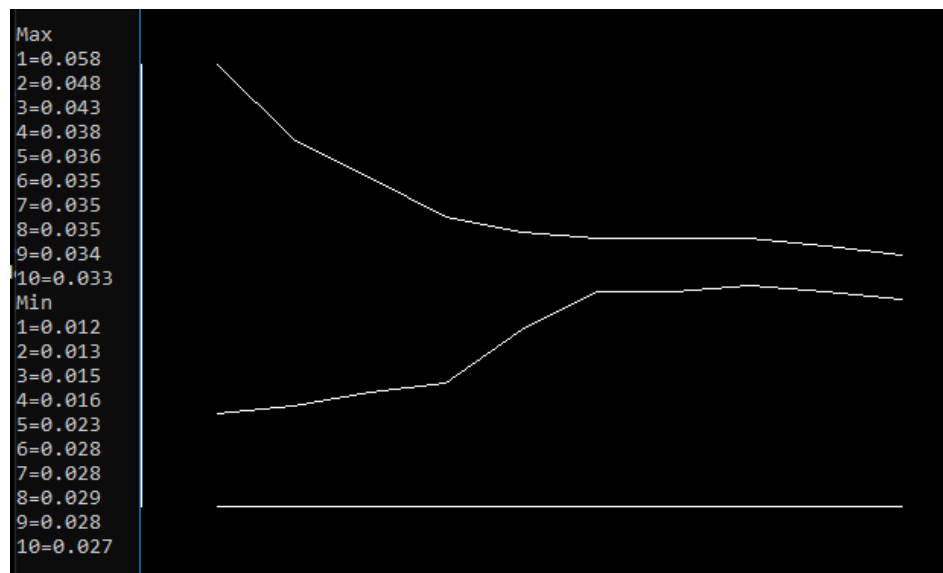
## Resultados:

En esta práctica se pide generación de 5, 10 Y 15 generaciones, les mostrare los máximos y mínimos de cada prueba y la gráfica.

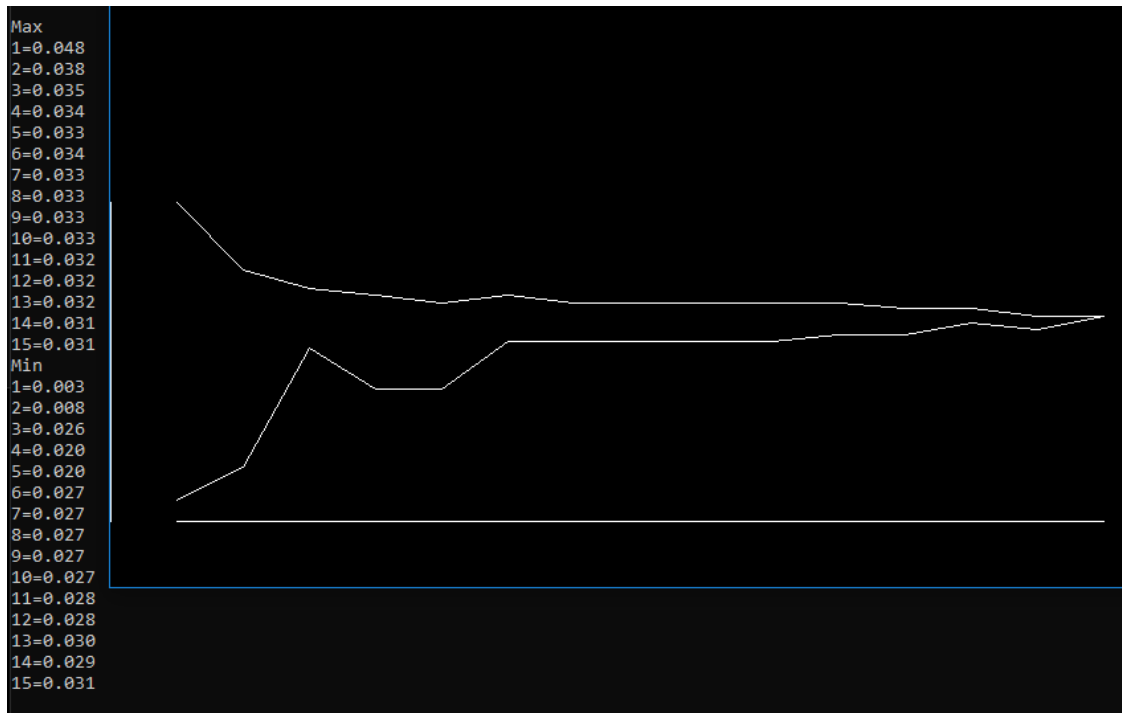
Para 5 generaciones:



Para 10 generaciones:



Para 15 generaciones:



## Conclusiones:

Como sabemos en el pasado este era uno de los algoritmos de selección mas utilizados, pero ahora es muy poco utilizado y analizando la gráfica creo que me puedo dar una idea, e 15 generaciones en una cuantas mas el algoritmo estaba a nada de convergir ya que al haber mas numero individuos que se acercan al máximo el máximo va disminuyendo,

Bueno en las tres graficas se nota un simple patrón que el máximo disminuye y el mínimo aumenta ya por lo que la convergencia ya no tardaba.