

# PRACTICA IV: SELECCIÓN MEDIANTE TORNEO

ALGORITMOS GENÉTICOS

3CM5

*Colín Varela, Alejandro*

## Contents

<b>1</b>	<b>Introducción</b>	<b>2</b>
<b>2</b>	<b>Desarrollo</b>	<b>2</b>
2.1	Código . . . . .	3
<b>3</b>	<b>Pruebas</b>	<b>7</b>
<b>4</b>	<b>Conclusiones</b>	<b>10</b>

## 1 Introducción

En esta practica programamos un Algoritmo genético basado en la selección por ruleta y gratificamos los por cientos de los máximos y mínimos para ver como se balancean entre si.

## 2 Desarrollo

Al igual que la práctica anterior, se opto por utilizar la librería 'plot-sdl' para poder graficar los máximos y mínimos de la aptitud y de las probabilidades. Lo que haremos ahora es mostrar el método de selección por ruleta para maximizar la función siguiente:

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

## 2.1 Código

Para esta practica se decidio re hacer el codigo que se tenia para hacer un algoritmo genetico de tal manera que fuera mucho mas sencillo adaptarlo a diferentes tipos de selecciones, aqui se muestra un boceto de como deberia quedar.

```
13  /*
14  |   Recibimos parametros:
15  |   - Numero de alelos
16  |   - Numero de Individuos
17  |   - Numero de Generaciones
18  |
19  |   Genera padres
20  |   for(Numero de generaciones)
21  |   {
22  |       Selecciona
23  |       Cruza
24  |       Mutacion
25  |   }
26  |
27  |   Grafica Ultima generaci3n
28  |   - Uno que grafique Maximos y minimos
29  |   - Otro que grafique probabilidades
30  |
31  |
32  |   main()
33  |   {
34  |       recibimos parametros();
35  |       generamos padres();
36  |       for(#generaciones)
37  |       {
38  |           selecciona("");
39  |           cruza();
40  |           muta():
41  |       }
42  |   }
43  */
```

En el main vamos a tener un ciclo por cada generación y pasaremos a los individuos por el método de selección de torneo (Que es el que intenta mostrar esta practica) seguido por la cruza, la muta al 30%, sacamos los máximos y los mínimos de esa generación, y finalmente imprimimos los resultados de esa generación. El proceso se repite por cada generación solicitada.

```
293     for(i=0;i<n_gen;i++)
294     {
295         seleccionTorneo();
296         cruza();
297         muta(30);
298         valuarInd();
299         maxMinProm(i);
300         maxMin(i);
301         printf("\nGeneracion %d :",i+1);
302         imprimir();
303     }
```

Para que pudiera funcionar de manera eficiente el programa esta vez se usaron varias funciones mostradas en la imagen:

```
50 void pedirDatos()
51 {
89 }
90 VAR decimal(VAR *in)
91 {
98 }
99 VAR funcionEv(VAR x)
100 {
104 }
105 void valuarInd()
106 {
117 }
118 void imprimir()
119 {
135 }
136 void generarPadres()
137 {
151 }
152 void descende(int x, int y)
153 {
157 }
158 void copiarDescendencia()
159 {
164 }
165 void seleccionTorneo()
166 {
223 }
224 void cruza()
225 {
238 }
239 void muta(int por)
240 {
249 }
250 void maxMinProm(int g)
251 {
263 }
264 void maxMin(int g)
265 {
277 }
```

Ahora sí veremos como funciona la función de la selección por torneo. Primero formamos las parejas que se van a comparar, revolviendo numeros del 1 al numero de individuos en un arreglo. Despues comparamos las parejas y descendemos al más fuerte.

```

165 void seleccionTorneo()
166 {
167     cnt3=0;
168     int cont=2;
169     //Barajar
170     int num;
171     int *b = (int*) malloc(n_ind * sizeof(int)),
172     *a = (int*) malloc(n_ind * sizeof(int));
173     while(cont>0)
174     {
175         for(cnt=0;cnt<n_ind;cnt++)
176         {
177             b[cnt]=1;
178         }
179         for(cnt=0;cnt<n_ind;cnt++)
180         {
181             do
182             {
183                 num = rand()%n_ind;
184             }
185             while(b[num]==0);
186             b[num]=0;
187             a[cnt]=num;
188         }
189         /*
190         for(cnt=0;cnt<n_ind;cnt++)
191         {
192             printf("%d | %d\n",b[cnt],a[cnt] +1);
193         }
194         */
195         //Seleccionamos padres
196         for(cnt=0;cnt<n_ind;cnt+=2)
197         {
198             num = rand()%101;
199             if(valor[a[cnt]]>valor[a[cnt+1]])
200             {
201                 if(num>30)
202                 {
203                     desciende(cnt3,a[cnt]);
204                 }
205                 else
206                     desciende(cnt3,a[cnt+1]);
207             }
208             else
209             {
210                 if(num>30)
211                 {
212                     desciende(cnt3,a[cnt+1]);
213                 }
214                 else
215                     desciende(cnt3,a[cnt]);
216             }
217             cnt3++;
218         }
219         cont--;
220     }
221     copiarDescendencia();
222     free(a);free(b);
223 }

```

Aquí también se muestra la función de aptitud transcrita en c:

```
VAR funcionEv(VAR x)
{
    VAR fx;
    fx = abs((x-5)/(2+sin(x)));
    return fx;
}
```

Para terminar simplemente enviamos los valores acumulados a la biblioteca para que lo pueda graficar.

```
//Graficamos
int ret = plot_graph(&params);
if (ret == EXIT_FAILURE)
{
    printf("plot_graph return with status %d\n",ret);
    return EXIT_FAILURE;
}
//~~~~ Grafica de valores ~~~~
printf("Se estan visualizando los : Valores\nCierra la ventana para salir\n\n");
//Creamos y populamos coordenadas
captionlist caption_list2 = NULL;
caption_list2=push_back_caption(caption_list2,"Maximos Valores",0,0x0000FF);
caption_list2=push_back_caption(caption_list2,"Minimos Valores",1,0xFF00FF);
coordlist coordinate_list2=NULL;
coordinate_list2=push_back_coord(coordinate_list2,0,0,0);
coordinate_list2=push_back_coord(coordinate_list2,1,0,0);
//Populamos
for(cnt=1;cnt<=n_gen;cnt++)
{
    coordinate_list2=push_back_coord(coordinate_list2,0,cnt,max[cnt-1]);
    coordinate_list2=push_back_coord(coordinate_list2,1,cnt,min[cnt-1]);
}
//Parametros
params.plot_window_title="Valores de Aptitud de las Generaciones";
params.scale_y = 1;
params.max_y = max[n_gen-1]+10;
params.caption_list = caption_list2;
params.coordinate_list = coordinate_list2;
//Graaficamos
ret = plot_graph(&params);
if (ret == EXIT_FAILURE)
{
    printf("plot_graph return with status %d\n",ret);
    return EXIT_FAILURE;
}
```

Nota: Favor de leer el README.txt para mas información sobre como compilar y correr la practica

### 3 Pruebas

Primero abrimos una consola en la carpeta de la practica y compilamos con 'make' , nos debe mostrar esta salida para una compilación exitosa. Posteriormente corremos con el comando descrito en el README.txt y nos sale el menu, para este ejemplo se usan 16 individuos y 30 generaciones.

```
Terminal - colin@Manjaro-de-Colin-42:~/Documents/Escuela/Algoritmos Geneticos/Practica 4 AG Colin_Vare
File Edit View Terminal Tabs Help

Generacion 1 :
N°      Individuo      Valor decimal      Aptitud      Promedio
1       1 1 0 1         13                 3.00         9.09
2       0 1 0 0         4                  0.00         0.00
3       1 1 1 1         15                 3.00         9.09
4       1 0 0 1         9                  1.00         3.03
5       0 0 0 0         0                  2.00         6.06
6       0 0 0 0         0                  2.00         6.06
7       1 0 1 1         11                 5.00         15.15
8       0 0 0 0         0                  2.00         6.06
9       1 0 0 0         8                  1.00         3.03
10      0 0 0 1         1                  1.00         3.03
11      0 1 1 0         6                  0.00         0.00
12      1 0 1 1         11                 5.00         15.15
13      0 0 0 1         1                  1.00         3.03
14      1 1 1 1         15                 3.00         9.09
15      0 0 0 1         1                  1.00         3.03
16      1 1 1 1         15                 3.00         9.09

Generacion 2 :
N°      Individuo      Valor decimal      Aptitud      Promedio
1       1 0 0 0         8                  1.00         3.45
```



Aquí vemos la ultima generación y también vemos el cartel que nos indica que se ha creado la gráfica de promedios y se esta visualizando

Terminal - colin@Manjaro-de-Colin-42:~/Documents/Escuela/Algoritmos Geneticos/Practica 4 AG Colin\_Vare

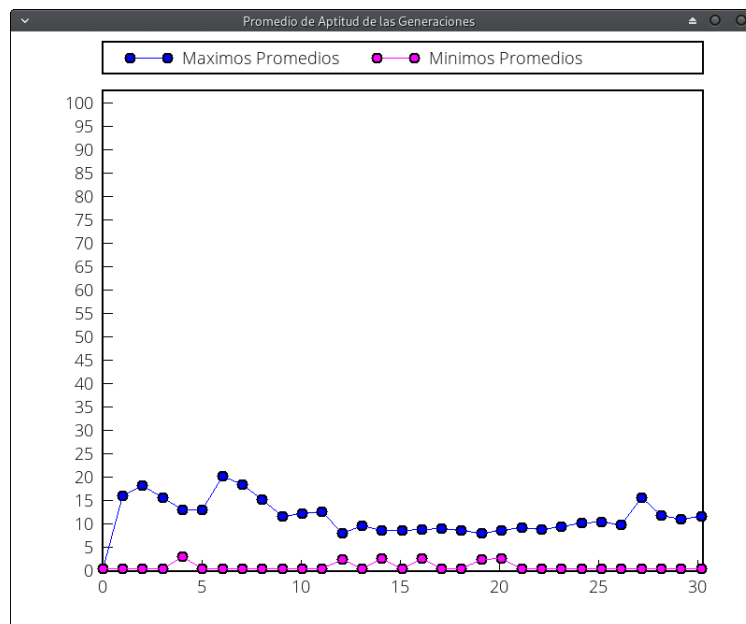
File Edit View Terminal Tabs Help

Generacion 30 :

N°	Individuo	Valor decimal	Aptitud	Promedio
1	1 1 0 0	12	4.00	10.81
2	1 1 0 1	13	3.00	8.11
3	0 1 0 0	4	0.00	0.00
4	0 1 0 0	4	0.00	0.00
5	1 1 0 0	12	4.00	10.81
6	0 1 0 1	5	0.00	0.00
7	1 1 1 0	14	3.00	8.11
8	1 0 0 0	8	1.00	2.70
9	1 1 0 0	12	4.00	10.81
10	0 0 0 1	1	1.00	2.70
11	0 1 1 0	6	0.00	0.00
12	1 1 0 0	12	4.00	10.81
13	1 1 0 0	12	4.00	10.81
14	1 1 0 1	13	3.00	8.11
15	1 1 1 1	15	3.00	8.11
16	1 0 1 0	10	3.00	8.11

Se estan visualizando los : Promedios  
Cierra la ventana para ver los valores

Aquí esta la grafica de promedios



Al cerrar esta grafica se nos indicara que ahora se visualizara la grafica de máximos y mínimos con respecto a los valores.

```

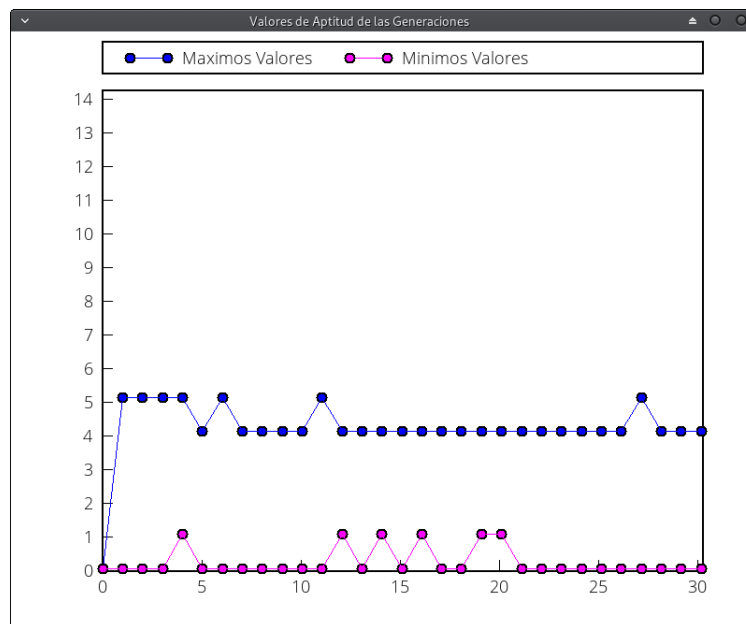
Terminal - colin@Manjaro-de-Colin-42:~/Documents/Escuela/Algoritmos Geneticos/Practica 4 AG Colin_Vare
File Edit View Terminal Tabs Help
1      | 1 1 0 0      |      12      | 4.00      | 10.81
2      | 1 1 0 1      |      13      | 3.00      | 8.11
3      | 0 1 0 0      |      4       | 0.00      | 0.00
4      | 0 1 0 0      |      4       | 0.00      | 0.00
5      | 1 1 0 0      |      12      | 4.00      | 10.81
6      | 0 1 0 1      |      5       | 0.00      | 0.00
7      | 1 1 1 0      |      14      | 3.00      | 8.11
8      | 1 0 0 0      |      8       | 1.00      | 2.70
9      | 1 1 0 0      |      12      | 4.00      | 10.81
10     | 0 0 0 1      |      1       | 1.00      | 2.70
11     | 0 1 1 0      |      6       | 0.00      | 0.00
12     | 1 1 0 0      |      12      | 4.00      | 10.81
13     | 1 1 0 0      |      12      | 4.00      | 10.81
14     | 1 1 0 1      |      13      | 3.00      | 8.11
15     | 1 1 1 1      |      15      | 3.00      | 8.11
16     | 1 0 1 0      |      10      | 3.00      | 8.11

Se estan visualizando los : Promedios
Cierra la ventana para ver los valores

Se estan visualizando los : Valores
Cierra la ventana para salir

```

Y aquí se encuentra dicha grafica de los valores



## 4 Conclusiones

Esta vez cambiamos a un método de selección muy distinto, este es el método de selección por torneo que desde un inicio me recordó a la ley del más fuerte. con el paso de la practica descubrí que mi idea no era tan errada y que con el método podíamos tomar a los individuos y asegurarnos de que solo desciendan los individuos con una mejor aptitud (o fuerza)