

Instituto Politécnico Nacional

Escuela Superior de Cómputo

Genetic Algorithms

Reporte Tercera Práctica:
“Cruza de individuos por el método de la ruleta”

Grupo 3CM5
Romero Godinez José Enrique

Introducción:

El método de selección de la ruleta consiste en hacer tantas iteraciones como el tamaño la población para seleccionar a los individuos “más aptos” dependiendo de su probabilidad de ser escogido al darle la vuelta la “ruleta” la cual es la representación de la distribución de probabilidad de la población.

Aunque es el método más utilizado, no es el mejor, ya que cabe la posibilidad de que escojamos individuos con un valor de aptitud muy bajo en cada iteración de la ruleta, llenando nuestra población de individuos poco aptos.[1]

El método de cruce por punto de cruce consiste en intercambiar determinados alelos desde un punto en común entre los padres para dar paso a su descendencia que tiene estas características.

El método de mutación de cambio de un bit, permite cambiar el valor de un alelo aleatorio en un cierto porcentaje de la población, este cambio puede volver al individuo más o menos apto, según sea el caso.

Objetivo:

Desarrollar un programa en lenguaje c que haga un experimento de selección con 5,10 y 15 generaciones de individuos utilizando el método de selección por ruleta, cruce por punto de cruce y mutación de cambio de bit al 10% de la población.

Solución:

Dado que el lenguaje c no permite de manera nativa las interfaces gráficas se hará uso del programa gnuplot[2] de linux, el cuál será invocado por un programa escrito en lenguaje c que será el encargado de llevar a cabo la selección, cruce y mutación de los individuos para finalmente graficar a los individuos más y menos aptos de cada generación.

Implementación:

Código del programa en c:

```
#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <math.h>

void generaIndividuos(unsigned char*);

void seleccionaPadres(unsigned char*,float*);

void calculaDatos(unsigned char*,int*,float*);

int dimeMaximo(unsigned char*);

int dimeMinimo(unsigned char*);

void cruzar(unsigned char*);

void mutar(unsigned char*);

int main()
```

```

{
    unsigned char* individuos=(unsigned char*)malloc(sizeof(unsigned char)*32);
    int* aptitud=(int*)malloc(sizeof(int)*32);
    float* probabilidad=(float*)malloc(sizeof(float)*32);
    int mayor,menor;
    float mayores[10];
    float menores[10];
    generaIndividuos(individuos);
    FILE * archivoMaximos = fopen("puntosMaximos.txt", "w");
    FILE * archivoMinimos = fopen("puntosMinimos.txt", "w");
    for(int i=0;i<10;i++)
    {
        calculaDatos(individuos,aptitud,probabilidad);
        mayor=dimeMaximo(individuos);
        mayores[i]=probabilidad[mayor];
        menor=dimeMinimo(individuos);
        menores[i]=probabilidad[menor];
        //Aqui se guardan los datos a graficar
        fprintf(archivoMaximos, "%d %f \n",i+1,mayores[i]);
        fprintf(archivoMinimos, "%d %f\n",i+1,menores[i]);
        //Guardar datos a graficar
        seleccionaPadres(individuos,probabilidad);
        cruzar(individuos);
        mutar(individuos);
    }
    char * configGnuplot[] = {"set title \"Histograma\"",
        "set ylabel \"----Aptitud--->\"",
        "set xlabel \"----Generaciones--->\"",
        "set style data histogram",

```

```

        "set style histogram cluster gap 1",
        "plot \"puntosMaximos.txt\" using 1:2 with linespoints ,\"puntosMinimos.txt\"
using 1:2 with linespoints"
    };

```

```

FILE * ventanaGnuplot = popen ("gnuplot -persist", "w");
// Ejecuta los comandos de configGnuPlot 1 por 1
for (int k=0;k<6;k++){
    fprintf(ventanaGnuplot, "%s \n", configGnuplot[k]);
}
fclose(archivoMinimos);
fclose(archivoMaximos);
//free(individuos);
free(aptitud);
free(probabilidad);

    return 0;
}

void calculaDatos(unsigned char* individuos,int* aptitud,float* probabilidad)
{
    float suma=0;
    for(int i=0;i<32;i++)
    {
        aptitud[i]=individuos[i]*individuos[i];
        suma+=(float)aptitud[i];
    }

    for(int i=0;i<32;i++)
    {
        probabilidad[i]=((float)aptitud[i]/suma);
    }
}

```

```

    }
}

void seleccionaPadres(unsigned char* individuos, float* probabilidad)
{
    unsigned char* seleccion=(unsigned char*)malloc(sizeof(unsigned char)*32);
    long ltime;
    double valor;
    float suma;
    ltime=time(NULL);
    int a=0,stime;
    stime=(unsigned)ltime/2;
    srand48(stime);
    for(int i=0;i<32;i++)
    {
        valor=drand48();
        while(suma<valor)
        {
            suma+=probabilidad[a];
            a++;
        }
        seleccion[i]=individuos[a];
        a=0;
        suma=0;
    }
    free(individuos);
    individuos=seleccion;
}

```

```

void generaIndividuos(unsigned char* individuos)

```

```

{
    long ltime;
    ltime=time(NULL);
    int a,stime;
    stime=(unsigned)ltime/2;
    srand(stime);
    for(int i=0;i<32;i++)
    {
        individuos[i]=(unsigned char)(1+rand()%(32-1));
    }
}

```

```

int dimeMaximo(unsigned char* individuos)

```

```

{
    int aux=0;
    for(int i=1;i<32;i++)
    {
        if(individuos[aux]<individuos[i])
        {
            aux=i;
        }
    }
    return aux;
}

```

```

int dimeMinimo(unsigned char* individuos)

```

```

{
    int aux=0;
    for(int i=1;i<32;i++)
    {
        if(individuos[aux]>individuos[i])

```

```

        {
            aux=i;
        }
    }
    return aux;
}

void cruzar(unsigned char* individuos)
{
    long ltime;
    ltime=time(NULL);
    int stime;
    stime=(unsigned)ltime/2;
    srand(stime);
    int puntoCruza=0;
    unsigned char* descendencia=(unsigned char*)malloc(sizeof(unsigned char)*32);
    int hijo=0;
    unsigned char bits,valor;
    puntoCruza=rand()%(5);
    for(int i=0;i<32;i++)
    {
        bits=((unsigned char)(pow((double)2,(double)puntoCruza)));
        if(hijo==0)
        {
            valor=bits&individuos[i];
            descendencia[i]=(individuos[i+1]-bits)+valor;
            hijo++;
        }
        if(hijo==1)
        {

```

```

        valor=bits&individuos[i];

        descendencia[i]=(individuos[i-1]-bits)+valor;

        hijo=0;

        puntoCruza=rand()%(5);

    }

}

free(individuos);

individuos=descendencia;

}

void mutar(unsigned char* individuos)
{
    long ltime;
    unsigned char bits,valor;
    int stime;
    int mutados[3],puntoCruza;

    ltime=time(NULL);
    stime=(unsigned)ltime/2;
    srand(stime);
    mutados[0]=rand()%(31);

    mutados[1]=rand()%(31);

    mutados[2]=rand()%(31);

    puntoCruza=rand()%(5);
    bits=((unsigned char)(pow(((double)2,((double)puntoCruza)))));
    for(int i=0;i<3;i++)
    {

```



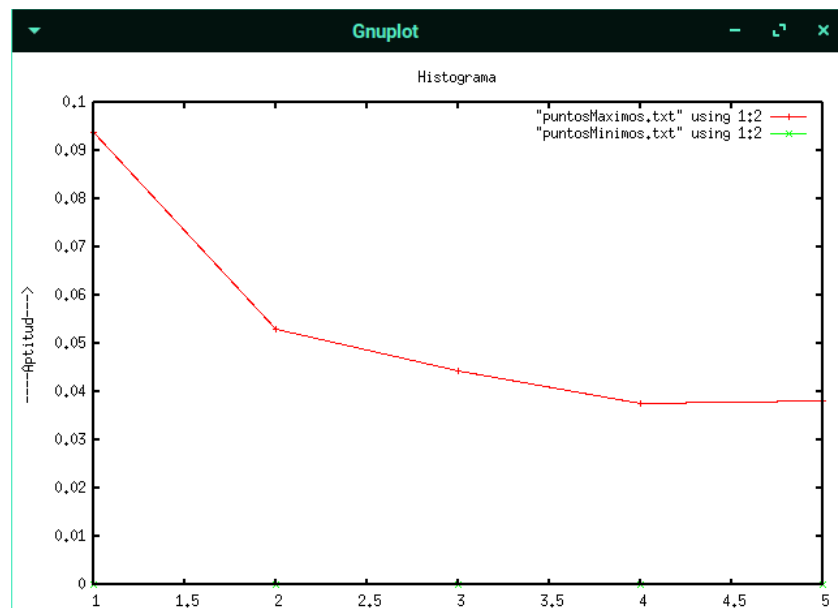
```

valor=bits&individuos[mutados[i]];
if(valor==0)
{
    individuos[mutados[i]]=individuos[mutados[i]]+bits;
}else{
    individuos[mutados[i]]=individuos[mutados[i]]-bits;
}
}
}

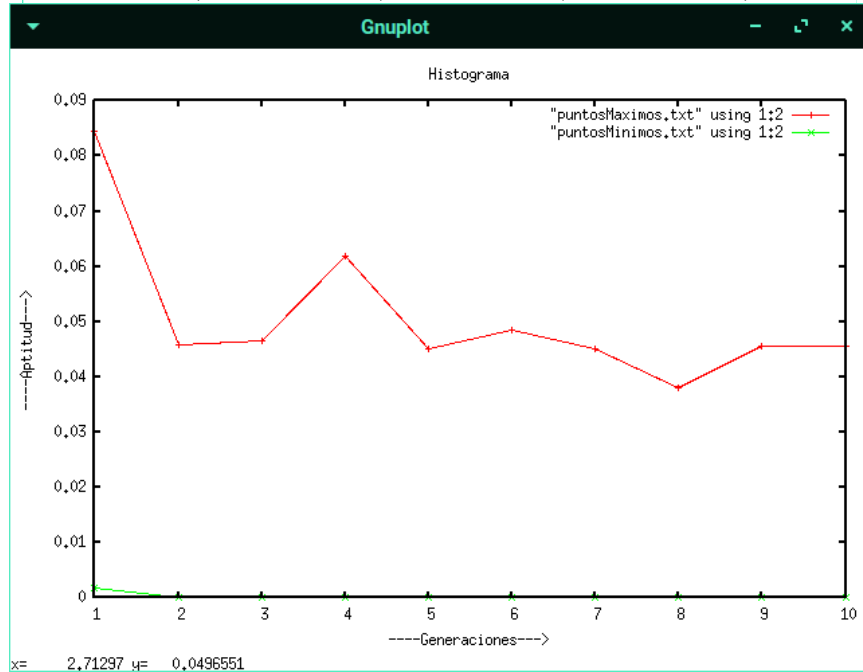
```

Ejecucion del programa:

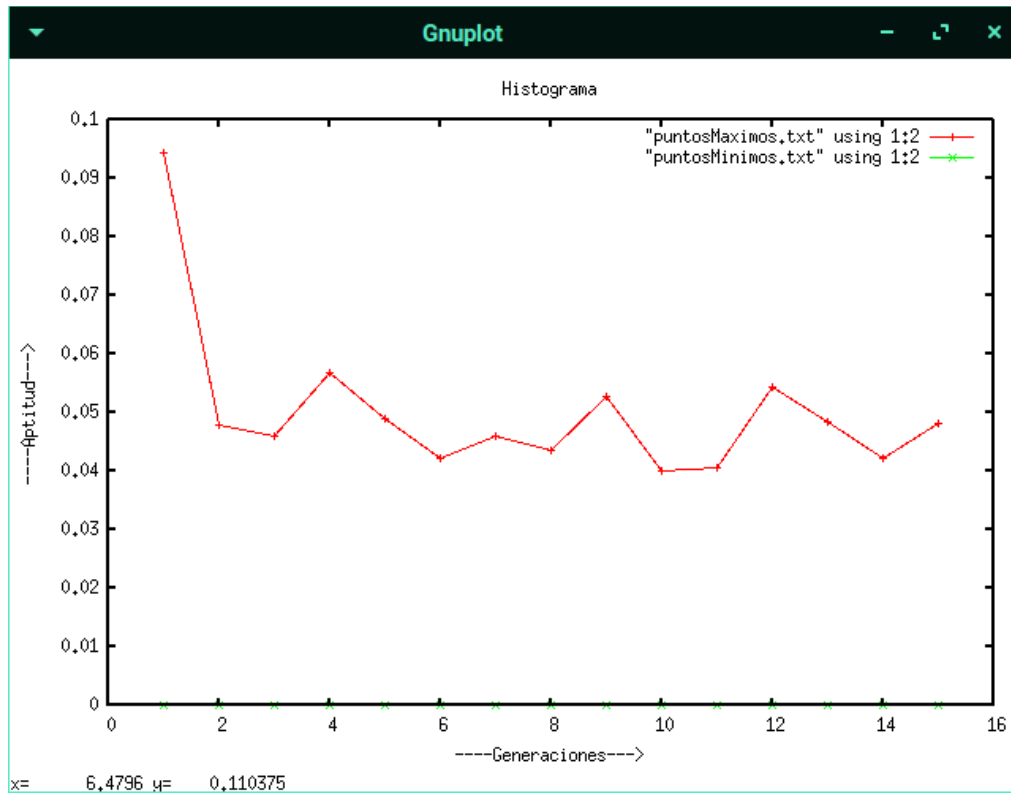
Experimento con 5 generaciones de individuos



Experimento con
10 generaciones:



Experimento con 15 generaciones:



Conclusiones:

El método de selección por ruleta es muy fácil de aplicar, pero llega a haber ciertos puntos en su ejecución donde los individuos menos aptos son escogidos, generando un cambio drástico en los valores de aptitud de la siguiente generación por lo que no es el mejor método a utilizar si lo que queremos es que nuestra población converja a un valor de aptitud alto.

Referencias:

[1]sabia.tic.udc.es/mgestal/cv/AAGGtutorial/node10.html

[2][https://www.google.com/url?](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=2ahUKEwiA1pLduqzdAhUPPq0KHXYUC3gQFjAAegQICRAB&url=http%3A%2F%2Fwww.gnuplot.info%2F&usg=AOvVaw3VAHCVnADPJiSQMpRM9jNG)

[sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=2ahUKEwiA1pLduqzdAhUPPq0KHXYUC3gQFjAAegQICRAB&url=http%3A%2F%2Fwww.gnuplot.info%2F&usg=AOvVaw3VAHCVnADPJiSQMpRM9jNG](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=2ahUKEwiA1pLduqzdAhUPPq0KHXYUC3gQFjAAegQICRAB&url=http%3A%2F%2Fwww.gnuplot.info%2F&usg=AOvVaw3VAHCVnADPJiSQMpRM9jNG)