

Instituto Politécnico Nacional

Escuela Superior de Cómputo

Genetic Algorithms

Reporte Sexta Práctica:
“Algoritmos de cruza Pt1”

Grupo 3CM5
Romero Godinez José Enrique

Introducción:

Existen diversos algoritmos de cruce que son utilizados, en esta parte analizaremos 4 de estos expuestos en los apuntes del Dr. Coello.

El primero de ellos es la cruce de un punto, la cual hemos ido utilizando en las practicas anteriores, este algoritmo consiste en fijar un punto en los alelos de cada individuo para proceder a la cruce intercambiando los cromosomas de los padres desde el punto definido, este algoritmo tiene como desventaja que puede que los cromosomas sean idénticos en el punto definido provocando que la descendencia no se vea modificada.

El segundo algoritmo es la cruce de dos puntos, en funcionamiento es exactamente igual que la cruce de un punto, sólo que ahora se seleccionará un punto adicional donde se “partirá” a los padres para intercambiar esta sección de cromosomas para los hijos.

El tercero es la cruce uniforme, consiste en seleccionar al azar puntos de cruce, que pueden ser vistos como una máscara de copiado, este algoritmo tiende a tener resultados disruptivos en la descendencia.

El cuarto y último algoritmo es la cruce acentuada, aquí se tomarán varios puntos de cruce de manera aleatoria, estos puntos indicarán de qué padre serán copiados los cromosomas, estos se comienzan a copiar 1 a 1 a los hijos, en el momento que se encuentra con un “acento” se procede a intercambiar el padre de donde se está copiando y viceversa.[1]

Objetivo:

Desarrollar un programa en lenguaje c que haga un experimento de cruce con los diferentes algoritmos expuestos en esta práctica.

Solución:

Se implementarán los algoritmos en lenguaje c, tomando en cuenta los pasos descritos en los apuntes del Dr. Coello para una población de 6 individuos de 8 cromosomas (caso particular, pueden ser n individuos de m cromosomas) .

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*,float*,float*);
```

```
void cruzaDeUnPunto(unsigned char*);
```

```
void cruzaDeDosPuntos(unsigned char*);
```

```
void cruzaUniforme(unsigned char*);
```

```
void cruzaAcentuada(unsigned char*);
void convierteBinario(unsigned char);
int main()
{
    unsigned char* individuos=(unsigned char*)malloc(sizeof(unsigned char)*6);
    int opc=0;
    generaIndividuos(individuos);
    printf("Selecciona una opcion de cruza\n1.-Cruza de un Punto\n2.-Cruza de Dos Puntos\n3.-
Cruza Uniforme\n4.-Cruza Acentuada\n");
    scanf("%d",&opc);
    switch(opc)
    {
        case 1:
            cruzaDeUnPunto(individuos);
            break;
        case 2:
            cruzaDeDosPuntos(individuos);
            break;
        case 3:
            cruzaUniforme(individuos);
            break;
        case 4:
            cruzaAcentuada(individuos);
            break;
        default:
            printf("Selecciona una opción valida\n");
            break;
    }
    free(individuos);
}
```

```

        return 0;
    }
void generaIndividuos(unsigned char* individuos)
{
    long ltime;
    ltime=time(NULL);
    int a,stime;
    stime=(unsigned)ltime/2;
    srand(stime);
    for(int i=0;i<6;i++)
    {
        individuos[i]=(unsigned char)(1+rand()%(255-1));
    }
}

```

```

void cruzaDeUnPunto(unsigned char* individuos)
{
    long ltime;
    ltime=time(NULL);
    int stime;
    stime=(unsigned)ltime/2;
    srand(stime);
    int puntoCruza=0;
    unsigned char bits,valor1,valor2,aux=0,hijo1,hijo2;
    for(int i=0;i<6;i=i+2)
    {
        aux=0;
        puntoCruza=rand()%(7);
        printf("Punto de Cruza: %d\n",puntoCruza);
    }
}

```

```

printf("Padre 1: ");
convierteBinario(individuos[i]);
printf("Padre 2: ");
convierteBinario(individuos[i+1]);
for(int j=7;j>7-puntoCruza;j--)
{
    aux+=((unsigned char)(pow((double)2,(double)j)));
}
//aux=aux+1;
//printf("Aux: %d\n",aux);

valor1=aux&individuos[i];
valor2=aux&individuos[i+1];
/*printf("Bits del padre 1: %d\n",valor1);
printf("Bits del padre 2: %d\n",valor2);*/
hijo1=(individuos[i+1]-(valor2))+valor1;
printf("Hijo 1: ");
convierteBinario(hijo1);
hijo2=(individuos[i]-(valor1))+valor2;
printf("Hijo 2: ");
convierteBinario(hijo2);
}
}

void cruzaDeDosPuntos(unsigned char* individuos)
{
    long ltime;
    ltime=time(NULL);
    int stime;

```

```

stime=(unsigned)ltime/2;
srand(stime);
int puntoCruza1=0,puntoCruza2=0;
unsigned char valor1,valor2,aux=0,hijo1,hijo2;
for(int i=0;i<6;i=i+2)
{
    aux=0;
    puntoCruza1=1+rand()%(7);
    puntoCruza2=rand()%(puntoCruza1-1);
    printf("Puntos de Cruza: %d,%d\n",puntoCruza1,puntoCruza2);
    printf("Padre 1: ");
    convierteBinario(individuos[i]);
    printf("Padre 2: ");
    convierteBinario(individuos[i+1]);
    for(int j=7-puntoCruza2;j>7-puntoCruza1;j--)
    {
        aux+=((unsigned char)(pow((double)2,(double)j)));
    }
    //aux=aux+1;
    printf("Aux: %d\n",aux);

    valor1=aux&individuos[i];
    valor2=aux&individuos[i+1];
    /*printf("Bits del padre 1: %d\n",valor1);
    printf("Bits del padre 2: %d\n",valor2);*/
    hijo1=(individuos[i+1]-(valor2))+valor1;
    printf("Hijo 1: ");
    convierteBinario(hijo1);
    hijo2=(individuos[i]-(valor1))+valor2;
}

```

```

        printf("Hijo 2: ");
        convierteBinario(hijo2);
    }

}

void cruzaUniforme(unsigned char* individuos)
{
    long ltime;
    ltime=time(NULL);
    int stime;
    stime=(unsigned)ltime/2;
    srand(stime);
    unsigned char mascara=0;
    unsigned char valor1,valor2,aux=0,hijo1,hijo2;
    for(int i=0; i<6;i=i+2)
    {
        for(int j=7;j>=0;j--)
        {
            mascara+=(rand()%2)*pow(2,j);
        }
        printf("Mascara de copiado: ");
        convierteBinario(mascara);
        printf("Padre 1: ");
        convierteBinario(individuos[i]);
        printf("Padre 2: ");
        convierteBinario(individuos[i+1]);
        valor1=mascara&individuos[i];
        valor2=mascara&individuos[i+1];
    }
}

```

```

        hijo1=(individuos[i+1]-valor2)+valor1;
        printf("Hijo 1: ");
        convierteBinario(hijo1);
        hijo2=(individuos[i]-(valor1))+valor2;
        printf("Hijo 2: ");
        convierteBinario(hijo2);
    }
}

void cruzaAcentuada(unsigned char* individuos)
{
    long ltime;
    ltime=time(NULL);
    int stime;
    stime=(unsigned)ltime/2;
    srand(stime);
    unsigned char mascara=0;
    unsigned char aux=0,hijo1=0,hijo2=0,padre=0;
    for(int i=0; i<6;i=i+2)
    {
        for(int j=7;j>=0;j--)
        {
            mascara+=(rand()%2)*pow(2,j);
        }
        printf("Acentos: ");
        convierteBinario(mascara);
        printf("Padre 1: ");
        convierteBinario(individuos[i]);
        printf("Padre 2: ");
        convierteBinario(individuos[i+1]);
    }
}

```



```

for(int j=7;j>=0;j--)
{
    if(mascara&(unsigned char)pow(2,j)){
        if(padre==0)
        {
            padre++;
        }else{
            padre=0;
        }
    }
    if(padre==0)
    {
        hijo1+=individuos[i]&(unsigned char)pow(2,j);
        hijo2+=individuos[i+1]&(unsigned char)pow(2,j);
    }else if(padre==1){
        hijo1+=individuos[i+1]&(unsigned char)pow(2,j);
        hijo2+=individuos[i]&(unsigned char)pow(2,j);
    }
}
printf("Hijo 1: ");
convierteBinario(hijo1);
printf("Hijo 2: ");
convierteBinario(hijo2);
hijo1=0;
hijo2=0;
}

}

void convierteBinario(unsigned char numero)

```

```

{
    unsigned char n=128;
    while(n!=0)
    {
        if(n&numero)
        {
            printf("1");
        }else{
            printf("0");
        }
        n=n/2;
    }
    printf("\n");
}

```

Ejecucion del programa:

Experimento con cruza de un punto:

```

Terminal - enrique@enrique-Notebook: ~/Documentos/Genetic algorithms/Practica 6
Archivo Editar Ver Terminal Pestañas Ayuda
enrique@enrique-Notebook:~/Documentos/Genetic algorithms/Practica 6$ ./a.out
Selecciona una opcion de cruza
1.-Cruza de un Punto
2.-Cruza de Dos Puntos
3.-Cruza Uniforme
4.-Cruza Acentuada
1
Punto de Cruza: 1
Padre 1: 11100001
Padre 2: 10101110
Hijo 1: 10101110
Hijo 2: 11100001
Punto de Cruza: 3
Padre 1: 10010000
Padre 2: 10101011
Hijo 1: 10001011
Hijo 2: 10110000
Punto de Cruza: 2
Padre 1: 00100110
Padre 2: 10100010
Hijo 1: 00100010
Hijo 2: 10100110
enrique@enrique-Notebook:~/Documentos/Genetic algorithms/Practica 6$

```

Experimento con cruza de 2 puntos:

```
Terminal - enrique@enrique-Notebook: ~/Documentos/Genetic algorithms/Practica 6
Archivo Editar Ver Terminal Pestañas Ayuda
1.-Cruza de un Punto
2.-Cruza de Dos Puntos
3.-Cruza Uniforme
4.-Cruza Acentuada
2
Puntos de Cruza: 4,1
Padre 1: 11011100
Padre 2: 10000111
Aux: 112
Hijo 1: 11010111
Hijo 2: 10001100
Puntos de Cruza: 6,3
Padre 1: 01101110
Padre 2: 11101111
Aux: 28
Hijo 1: 11101111
Hijo 2: 01101110
Puntos de Cruza: 7,3
Padre 1: 11110010
Padre 2: 01101000
Aux: 30
Hijo 1: 01110010
Hijo 2: 11101000
enrique@enrique-Notebook:~/Documentos/Genetic algorithms/Practica 6$
```

Experimento con cruza uniforme:

```
Terminal - enrique@enrique-Notebook: ~/Documentos/Genetic algorithms/Practica 6
Archivo Editar Ver Terminal Pestañas Ayuda
enrique@enrique-Notebook:~/Documentos/Genetic algorithms/Practica 6$ ./a.out
Selecciona una opcion de cruza
1.-Cruza de un Punto
2.-Cruza de Dos Puntos
3.-Cruza Uniforme
4.-Cruza Acentuada
3
Mascara de copiado: 00110111
Padre 1: 11100000
Padre 2: 10000110
Hijo 1: 10100000
Hijo 2: 11000110
Mascara de copiado: 01111010
Padre 1: 01110000
Padre 2: 10110010
Hijo 1: 11110000
Hijo 2: 00110010
Mascara de copiado: 11001010
Padre 1: 00010011
Padre 2: 10001011
Hijo 1: 00000011
Hijo 2: 10011011
enrique@enrique-Notebook:~/Documentos/Genetic algorithms/Practica 6$
```

Experimento con cruza acentuada:

```
Terminal - enrique@enrique-Notebook: ~/Documentos/Genetic algorithms/Practica 6
Archivo  Editar  Ver  Terminal  Pestañas  Ayuda
enrique@enrique-Notebook:~/Documentos/Genetic algorithms/Practica 6$ ./a.out
Selecciona una opcion de cruza
1.-Cruza de un Punto
2.-Cruza de Dos Puntos
3.-Cruza Uniforme
4.-Cruza Acentuada
4
Acentos: 01011111
Padre 1: 10001010
Padre 2: 10100100
Hijo 1: 10100000
Hijo 2: 10001110
Acentos: 11010001
Padre 1: 10111010
Padre 2: 00011011
Hijo 1: 00111010
Hijo 2: 10011011
Acentos: 01000110
Padre 1: 00010010
Padre 2: 00101101
Hijo 1: 00101001
Hijo 2: 00010110
enrique@enrique-Notebook:~/Documentos/Genetic algorithms/Practica 6$
```

Conclusiones:

Estos algoritmos de cruza funcionan de una manera similar, tomando partes de cada padre y heredándolas a los hijos, difieren en los resultados y son óptimos dependiendo de la población que queramos obtener, por ejemplo si utilizamos la cruza uniforme es casi seguro que los hijos serán una mezcla total de los padres o puede que los puntos de cruza sean tales que la descendencia termine siendo copias exactas de estos, este último caso suele presentarse con menor frecuencia en los otros tres algoritmos, estas particularidades parecieran ser solucionadas con la cruza acentuada debido a que su copiado se hace alternativamente.

Referencias:

[1]Introducción al Cómputo Evolutivo, Coello