

PRACTICA II: REPRESENTAR INDIVIDUOS

ALGORITMOS GENÉTICOS

3CM5

Colín Varela

Contents

1	Introducción	2
2	Desarrollo	2
2.1	Representación Binaria	2
2.2	Códigos Gray	3
2.3	Números Reales	4
2.4	Números Enteros	4
3	Pruebas	5
3.1	Representación Binaria	6
3.2	Códigos Gray	7
3.3	Números Reales	8
3.4	Números Enteros	9
4	Conclusiones	9

1 Introducción

En esta practica vamos a probar las formas de representación más básicas de individuos, estas son la representación binaria, código de gray, en números reales, y en números enteros. Con esto podremos empezar a ver como se comportan los algoritmos genéticos.

2 Desarrollo

Se utilizó un menú que contenía las cuatro representaciones, para así tener un más fácil acceso a cada una.

```
do
{
    printf("\n\nPractica 2: Representación de algoritmos geneticos\n");
    printf("Alumno: Alejandro Colin Varela\n");
    printf("Menu:\n");
    printf("1) Representación binaria\n");
    printf("2) Códigos de Gray\n");
    printf("3) Codificación en números reales\n");
    printf("4) Codificación en números enteros\n");
    printf(": ");
    scanf("%d",&op);
}
```

Aquí la vista del menú y sus opciones, acomodado con printf's y un scanf para determinar que escoge el usuario. Veamos los códigos de las cuatro partes del menú.

2.1 Representación Binaria

```
//Representación binaria
if(op==1)
{
    int A[10][7];
    for ( i = 0; i < 10; ++i)
        for ( i2 = 0; i2 < 7; ++i2)
            A[i][i2] = rand()%2; //Genera los numeros
    for ( i = 0; i < 10; ++i)
    {
        printf("\nIndividuo %d\n", i);
        for ( i2 = 0; i2 < 7; ++i2)
            printf(" %d |", A[i][i2]); //Imprime
    }
    char c = getchar();
    printf("\n\nENTER para volver al menu\n");
    c = getchar();
}
```

Primero creamos un arreglo de 10x7 lo que representa 10 individuos con 7 alelos. Ahora dentro de dos for, los llenamos con rand()%2, esta sentencia genera 1's y 0's por lo que los alelos tendrán unos y ceros cumpliendo la representación binaria.

2.2 Códigos Gray

```
//Códigos de Gray
else if(op==2)
{
    int A[10][7];
    for ( i = 0; i < 10; ++i)
        for ( i2 = 0; i2 < 7; ++i2)
            A[i][i2] = rand()%2;    //Genera los numeros

    for ( i = 0; i < 10; ++i)
        for ( i2 = 6; i2 > 0; --i2)
        {
            if(A[i][i2]==A[i][i2-1])    //Genera el Gray
                A[i][i2] = 0;
            else
                A[i][i2] = 1;
        }
    for ( i = 0; i < 10; ++i)
    {
        printf("\nIndividuo %d\n", i);
        for ( i2 = 0; i2 < 7; ++i2)
            printf(" %d |", A[i][i2]);    //Imprime
    }
    char c = getchar();
    printf("\n\nENTER para volver al menu\n");
    c = getchar();
}
```

Para los códigos Gray, primero vamos a generar los números binarios dentro de el arreglo 10x7 (10 individuos, 7 alelos). Una vez obtenida la cadena binaria, vamos a aplicarle de derecha a izquierda una puerta XOR y finalmente así quedara la representación en código gray. Después imprimimos.

2.3 Números Reales

```
//Codificación en números reales
else if(op==3)
{
    float A[10][7];
    for ( i = 0; i < 10; ++i)
        for ( i2 = 0; i2 < 7; ++i2)
        {
            float rand2 = rand()%1000;
            A[i][i2] = rand2/100; //Genera los numeros
        }
    for ( i = 0; i < 10; ++i)
    {
        printf("\nIndividuo %d\n", i);
        for ( i2 = 0; i2 < 7; ++i2)
            printf(" %.2f |", A[i][i2]); //Imprime
    }
    char c = getchar();
    printf("\n\nENTER para volver al menu\n");
    c = getchar();
}
```

Ya que en esta representación nos pide que sean números reales tenemos que dividir la salida de `rand()` ya que esta nos da números enteros. Así que una vez que tiene el número lo divide en 100 y entra el número en el arreglo. Obteniendo así números reales.

2.4 Números Enteros

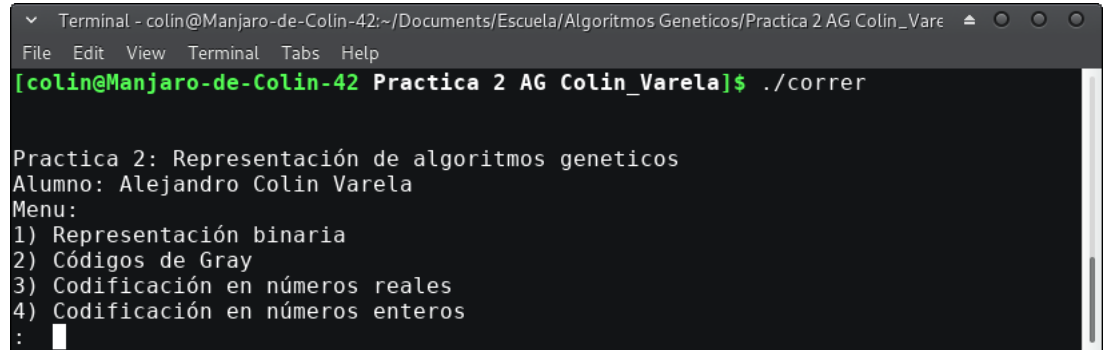
```
//Codificación en números enteros
else if(op==4)
{
    int A[10][7];
    for ( i = 0; i < 10; ++i)
        for ( i2 = 0; i2 < 7; ++i2)
            A[i][i2] = rand()%10; //Genera los numeros
    for ( i = 0; i < 10; ++i)
    {
        printf("\nIndividuo %d\n", i);
        for ( i2 = 0; i2 < 7; ++i2)
            printf(" %d |", A[i][i2]); //Imprime
    }
    char c = getchar();
    printf("\n\nENTER para volver al menu\n");
    c = getchar();
}
```

Esta representación resulto bastante sencilla ya que simplemente se debía hacer la representación con números enteros. Usamos `rand()` con un tope de 10, para efectos prácticos. Llenamos el arreglo e imprimimos.

3 Pruebas

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

Una vez compilado usando 'make' vamos a correr usando './correr' y nos debe aparecer el siguiente menú:



```
Terminal - colin@Manjaro-de-Colin-42:~/Documents/Escuela/Algoritmos Geneticos/Practica 2 AG Colin_Varela
File Edit View Terminal Tabs Help
[colin@Manjaro-de-Colin-42 Practica 2 AG Colin_Varela]$ ./correr

Practica 2: Representación de algoritmos geneticos
Alumno: Alejandro Colin Varela
Menu:
1) Representación binaria
2) Códigos de Gray
3) Codificación en números reales
4) Codificación en números enteros
: 
```

En este menú podemos escoger cualquiera de las cuatro representaciones.

3.1 Representación Binaria

```
Terminal - colin@Manjaro-de-Colin-42:~/Documents/Escuela/Algoritmos Geneticos/Practica 2 AG Colin_Varela
File Edit View Terminal Tabs Help
Practica 2: Representación de algoritmos geneticos
Alumno: Alejandro Colin Varela
Menu:
1) Representación binaria
2) Códigos de Gray
3) Codificación en números reales
4) Codificación en números enteros
: 1

Individuo 0
1 | 1 | 0 | 1 | 0 | 1 | 0 |
Individuo 1
0 | 1 | 1 | 0 | 1 | 1 | 1 |
Individuo 2
0 | 0 | 0 | 0 | 1 | 1 | 0 |
Individuo 3
1 | 0 | 0 | 1 | 1 | 0 | 1 |
Individuo 4
1 | 0 | 0 | 1 | 1 | 1 | 0 |
Individuo 5
1 | 0 | 0 | 1 | 1 | 1 | 0 |
Individuo 6
0 | 1 | 1 | 1 | 1 | 1 | 1 |
Individuo 7
0 | 0 | 0 | 1 | 1 | 0 | 0 |
Individuo 8
0 | 1 | 1 | 0 | 1 | 0 | 1 |
Individuo 9
0 | 1 | 1 | 1 | 1 | 1 | 1 |

ENTER para volver al menu
```

Podemos apreciar los 10 individuos y sus respectivos alelos que en este caso son números binarios aleatorios.

3.2 Códigos Gray

```
Terminal - colin@Manjaro-de-Colin-42:~/Documents/Escuela/Algoritmos Geneticos/Practica 2 AG Colin_Varela
File Edit View Terminal Tabs Help
Practica 2: Representación de algoritmos geneticos
Alumno: Alejandro Colin Varela
Menu:
1) Representación binaria
2) Códigos de Gray
3) Codificación en números reales
4) Codificación en números enteros
: 2

Individuo 0
0 | 0 | 1 | 0 | 0 | 1 | 0 |
Individuo 1
0 | 1 | 0 | 1 | 0 | 1 | 0 |
Individuo 2
1 | 1 | 1 | 0 | 0 | 0 | 0 |
Individuo 3
0 | 1 | 1 | 0 | 0 | 1 | 0 |
Individuo 4
1 | 1 | 0 | 1 | 0 | 0 | 1 |
Individuo 5
0 | 1 | 1 | 1 | 0 | 1 | 1 |
Individuo 6
1 | 0 | 0 | 1 | 1 | 1 | 1 |
Individuo 7
0 | 1 | 0 | 1 | 0 | 1 | 1 |
Individuo 8
0 | 1 | 1 | 1 | 0 | 1 | 1 |
Individuo 9
0 | 0 | 1 | 0 | 0 | 1 | 0 |

ENTER para volver al menu
```

Se imprimen los 10 individuos y se puede apreciar un patrón de código gray, si imprimiéramos la cadena antes de la XOR se vería como esta cambió por la misma.

3.3 Números Reales

```
Terminal - colin@Manjaro-de-Colin-42:~/Documents/Escuela/Algoritmos Geneticos/Practica 2 AG Colin_Varela
File Edit View Terminal Tabs Help

Practica 2: Representación de algoritmos geneticos
Alumno: Alejandro Colin Varela
Menu:
1) Representación binaria
2) Códigos de Gray
3) Codificación en números reales
4) Codificación en números enteros
: 3

Individuo 0
9.10 | 7.46 | 9.95 | 9.81 | 3.23 | 7.80 | 4.95 |
Individuo 1
1.97 | 9.93 | 4.65 | 2.99 | 2.94 | 5.48 | 6.98 |
Individuo 2
6.85 | 3.15 | 1.98 | 7.75 | 6.66 | 3.52 | 5.71 |
Individuo 3
1.16 | 8.35 | 3.34 | 7.42 | 9.63 | 4.51 | 4.59 |
Individuo 4
1.56 | 0.55 | 2.69 | 0.67 | 8.01 | 2.65 | 0.48 |
Individuo 5
1.25 | 3.97 | 8.96 | 3.22 | 7.42 | 3.61 | 9.73 |
Individuo 6
3.89 | 2.61 | 6.71 | 0.74 | 5.76 | 8.70 | 2.01 |
Individuo 7
5.95 | 5.74 | 1.24 | 0.63 | 7.61 | 8.10 | 1.57 |
Individuo 8
7.24 | 6.14 | 9.69 | 2.33 | 6.69 | 2.38 | 3.00 |
Individuo 9
4.71 | 8.55 | 7.00 | 9.48 | 2.53 | 5.96 | 6.22 |

ENTER para volver al menu
```

Podemos ver como fue un éxito el dividir el `rand()` entre 100 pues gracias a ello tenemos números flotantes en los alelos de los 10 individuos que se muestran.

3.4 Números Enteros

```
Terminal - colin@Manjaro-de-Colin-42:~/Documents/Escuela/Algoritmos Geneticos/Practica 2 AG Colin_Varela
File Edit View Terminal Tabs Help

Practica 2: Representación de algoritmos geneticos
Alumno: Alejandro Colin Varela
Menu:
1) Representación binaria
2) Códigos de Gray
3) Codificación en números reales
4) Codificación en números enteros
: 4

Individuo 0
7 | 7 | 5 | 6 | 0 | 7 | 2 |
Individuo 1
9 | 9 | 6 | 4 | 3 | 0 | 9 |
Individuo 2
7 | 3 | 6 | 3 | 7 | 5 | 6 |
Individuo 3
6 | 6 | 6 | 9 | 1 | 7 | 9 |
Individuo 4
6 | 5 | 1 | 4 | 5 | 7 | 2 |
Individuo 5
5 | 6 | 5 | 4 | 5 | 1 | 0 |
Individuo 6
0 | 3 | 9 | 9 | 6 | 8 | 3 |
Individuo 7
3 | 5 | 9 | 2 | 1 | 8 | 3 |
Individuo 8
5 | 7 | 3 | 1 | 4 | 4 | 7 |
Individuo 9
9 | 3 | 0 | 7 | 1 | 5 | 1 |

ENTER para volver al menu
```

En esta parte vemos que el tope de 10 funciona, y en los alelos de los 10 individuos hay números enteros y completamente aleatorios, así cumplimos las restricciones.

4 Conclusiones

Gracias a esta practica se pudo apreciar de mejor manera la forma que tienen los individuos y de que maneras podemos representarlos. Seguramente esto sera de gran ayuda combinado con lo siguiente del curso. Pues con ello podemos entender de una mejor manera los algoritmos genéticos.