“ESCUELA SUPERIOR DE CÓMPUTO”

MATERIA: GENETIC ALGORITHMS

PROFESOR: MORALES GUITRON SANDRA LUZ

GRUPO: 3CM5

ALUMNO: ALBARRAN CRUZ CARLOS ALBERTO

“PRACTICA 2”

INTRODUCCIÓN

El objetivo de esta práctica número 2 es la creación de individuos en sus diferentes representaciones.

- Representación Binaria.
  La representación usada por el algoritmo genético. La representación tradicional usada para codificar un conjunto de soluciones es el esquema binario en eel cual un cromosoma1 es una cadena de la forma (b1,b2,...,bm), donde b1,b2,..,bm se denominan alelos(ya sea ceros o unos). Hay varias razones por las cuales suele usarse la codificación binaria en los AGs, aunque la mayoría de ellas se remontan al trabajo pionera de Holland en el área.
- Códigos de Gray.
  Consiste en una ordenación de n^2 números binarios de tal forma que cada número sólo tenga un dígito binario distinto a su predecesor. Esta técnica de codificación se originó cuando los circuitos lógicos digitales se realizaban con válvulas de vacío y dispositivos electromecánicos. Los contadores necesitaban potencias muy elevadas a la entrada y generaban picos de ruido cuando varios bits cambian simultáneamente. El uso de código Gray garantizó que en cualquier transición variaría tan sólo un bit.
- Codificación en números reales.
  El uso directo de números reales en un cromosoma funciona mejor en la práctica que la representación binaria tradicional.
- Codificación en números enteros
  Una representación entera de números reales. La cadena completa es decodificada como un solo número era multiplicando y dividiendo cada dígito de acuerdo a su posición.

CONTENIDO

```cpp
#include <iostream>
#include <stdio.h>
#include <time.h>
#include <random> //Libreria para la generacion de numeros aleatorios

using namespace std;

void binario(int **); //PROTOTIPOS
void gray(int **);
void real(float **);
void entero(int **);
void despliega(int **, float **);


void main() {
    int opcion; //Inicializacion de arreglos para los individuos
    int** arreglo;
    arreglo = (int **)malloc(sizeof(int*)*10); //Se creean 10 individuos
    float** farreglo;
    farreglo = (float **)malloc(sizeof(float*) * 10);
    for (int i = 0; i < 10; i++) {
        farreglo[i] = (float*)malloc(sizeof(float) * 10); //Se crea cada individuo con 10 alelos
    }

    for (int i = 0; i < 10; i++) {
        arreglo[i] = (int*)malloc(sizeof(int) * 10);
    }
```

```cpp
    for (int i = 0; i < 10; i++) {
        arreglo[i] = (int*)malloc(sizeof(int) * 10);
    }

    cout << "1. BINARIO" << endl<<   //MENU DESPLEGABLE PARA SELECCIONAR EL TIPO DE REPRESENTACION
            "2. CODIGO GRAY"<<endl<<
        "3. NUMEROS REALES" << endl <<
        "4. NUMEROS ENTEROS" << endl <<
        "5. Salir"<<endl<<
        "Teclee la opcion deseada"<<endl;
    cin >> opcion;

    switch (opcion)
    {
    case 1:
        binario(arreglo); //1. Representacion binaria
        main();
        break;
    case 2:
        gray(arreglo); //2. Representacion con codigo gray
        main();
        break;
    case 3:
        real(farreglo); //3. Representación real
        main();
        break;
    case 4:
        entero(arreglo); //4. Representacion entera
        main();
        break;
    case 5:
        exit(0);
        break;
    default:
        cout << "Opcion incorrecta" << endl;
        exit(0);
```

```
        exit(0);
        break;
    }
}

void binario(int** arreglo) {
    std::default_random_engine generator; //Motor generador de numeros aleatorios
    std::uniform_int_distribution<int> distribution(0, 1); //Tipode distribucion para generar aleatorios

    for (int i=0;i<10;i++){
        for (int j = 0; j < 10; j++) {
            int dice_roll = distribution(generator);  // Se generan los numeros aleatorios
            arreglo[i][j] = dice_roll;                  //Se llena cada individuo

        }

    }

    despliega(arreglo,NULL); //Se despliegan los 10 individuos

}
void real(float** farreglo) {
    std::default_random_engine generator;
    std::uniform_real_distribution<float> distribution(1.00f, 100.00f);

    for (int i = 0; i < 10; i++) {
        for (int j = 0; j < 10; j++) {
            float dice_roll = distribution(generator);  // Genera numeros reales
            farreglo[i][j] = dice_roll;


        }
```

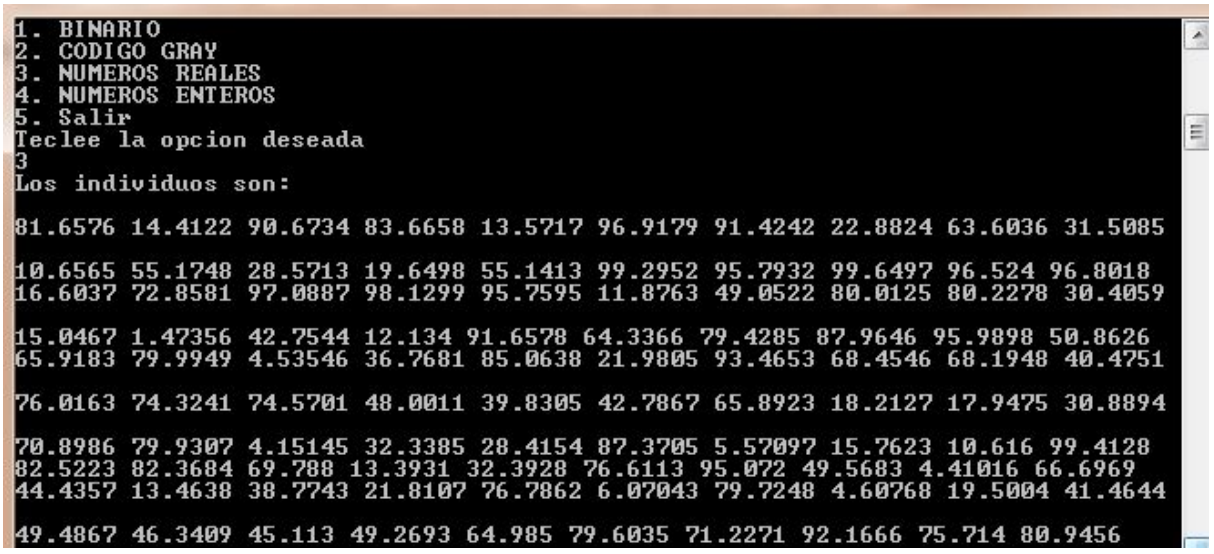
```
    }

    despliega(NULL, farreglo);
}
void entero(int** arreglo) {
    std::default_random_engine generator;
    std::uniform_int_distribution<int> distribution(0, 100);

    for (int i = 0; i < 10; i++) {
        for (int j = 0; j < 10; j++) {
            int dice_roll = distribution(generator);  // Se generan numeros 1-0
            arreglo[i][j] = dice_roll;

        }

    }

    despliega(arreglo, NULL);
}
void gray(int** arreglo) {
    std::default_random_engine generator;
    std::uniform_int_distribution<int> distribution(0, 1);

    for (int i = 0; i < 10; i++) {
        for (int j = 0; j < 10; j++) {
            int dice_roll = distribution(generator);  // Se generan numeros 1-0
            arreglo[i][j] = dice_roll;

        }

    }

    despliega(arreglo, NULL);

```

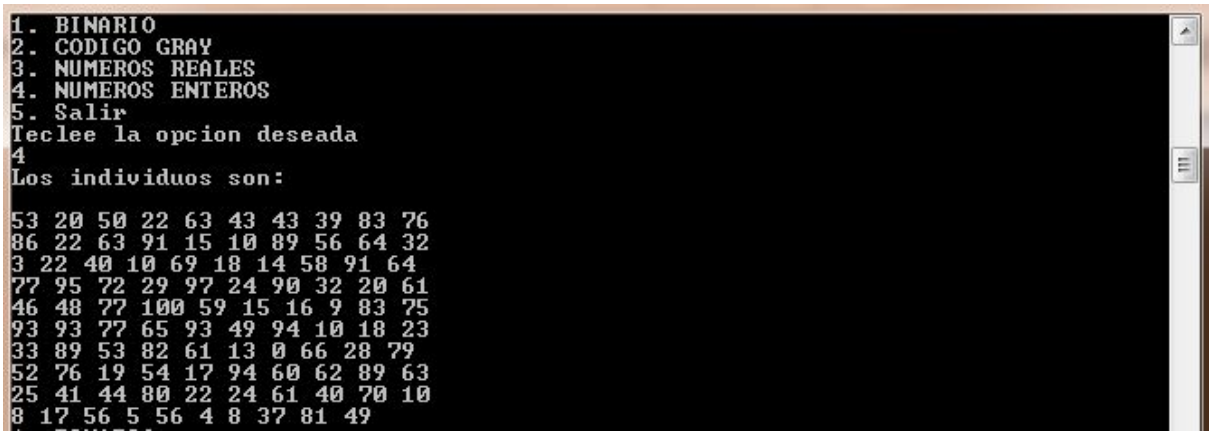
```
}

void despliega(int** arreglo, float** farreglo) { //Funcion para desplegar a los individuos
    cout << "Los individuos son:" << endl;
    if (arreglo != NULL) {
        for (int i = 0; i < 10; i++) {
            cout << endl;
            for (int j = 0; j < 10; j++) {
                cout << arreglo[i][j] << " ";

            }


        }
        free(arreglo);
        cout << endl;
    }
    else {
        for (int i = 0; i < 10; i++) {
            cout << endl;
            for (int j = 0; j < 10; j++) {
                cout << farreglo[i][j] << " ";

            }


        }
        free(farreglo);
        cout << endl;
    }


}
```

RESULTADOS


1. BINARIO
2. CODIGO GRAY
3. NUMEROS REALES
4. NUMEROS ENTEROS
5. Salir
Teclee la opcion deseada
1
Los individuos son:
0 0 0 1 0 1 1 1 0 1
1 1 0 0 1 0 1 0 1 0
1 1 1 1 0 0 0 1 0 1
0 1 1 0 0 1 1 1 1 0
0 0 1 0 1 0 1 0 1 0
0 1 0 0 0 0 0 1 0 0
1 1 0 0 0 0 0 0 1 1
1 0 1 1 1 0 1 0 0 0
0 1 1 0 0 1 0 1 0 1
1 0 0 1 1 0 0 1 0 0


1. BINARIO
2. CODIGO GRAY
3. NUMEROS REALES
4. NUMEROS ENTEROS
5. Salir
Teclee la opcion deseada
2
Los individuos son:
0 0 0 1 0 1 1 1 0 1
1 1 0 0 1 0 1 0 1 0
1 1 1 1 0 0 0 1 0 1
0 1 1 0 0 1 1 1 1 0
0 0 1 0 1 0 1 0 1 0
0 1 0 0 0 0 0 1 0 0
1 1 0 0 0 0 0 0 1 1
1 0 1 1 1 0 1 0 0 0
0 1 1 0 0 1 0 1 0 1
1 0 0 1 1 0 0 1 0 0


1. BINARIO
2. CODIGO GRAY
3. NUMEROS REALES
4. NUMEROS ENTEROS
5. Salir
Teclee la opcion deseada
3
Los individuos son:
81.6576 14.4122 90.6734 83.6658 13.5717 96.9179 91.4242 22.8824 63.6036 31.5085
10.6565 55.1748 28.5713 19.6498 55.1413 99.2952 95.7932 99.6497 96.524 96.8018
16.6037 72.8581 97.0887 98.1299 95.7595 11.8763 49.0522 80.0125 80.2278 30.4059
15.0467 1.47356 42.7544 12.134 91.6578 64.3366 79.4285 87.9646 95.9898 50.8626
65.9183 79.9949 4.53546 36.7681 85.0638 21.9805 93.4653 68.4546 68.1948 40.4751
76.0163 74.3241 74.5701 48.0011 39.8305 42.7867 65.8923 18.2127 17.9475 30.8894
70.8986 79.9307 4.15145 32.3385 28.4154 87.3705 5.57097 15.7623 10.616 99.4128
82.5223 82.3684 69.788 13.3931 32.3928 76.6113 95.072 49.5683 4.41016 66.6969
44.4357 13.4638 38.7743 21.8107 76.7862 6.07043 79.7248 4.60768 19.5004 41.4644
49.4867 46.3409 45.113 49.2693 64.985 79.6035 71.2271 92.1666 75.714 80.9456



CONCLUSIONES

La generación de varios individuos en este caso arreglos de enteros o flotantes sea el caso es un proceso que tiene bastante factores en que pensar y esta a disposicion del que haga los individuos de cuántos alelos estará hecho dada individuo, que representación usará o cuantos individuos generará.