

**INSTITUTO POLITÉCNICO  
NACIONAL**



**Escuela Superior de  
Computo**

# **ALGORITMOS GENÉTICOS**

**Práctica 2**

**3CM5**

**Fernando Zamora Galloso**

# INTRODUCCIÓN

## Algoritmos genéticos

En los años 1970, de la mano de John Henry Holland, surgió una de las líneas más prometedoras de la inteligencia artificial, la de los **algoritmos genéticos**, (AG). Son llamados así porque se inspiran en la evolución biológica y su base genético-molecular.

Estos algoritmos hacen evolucionar una población de individuos sometiéndola a acciones aleatorias semejantes a las que actúan en la evolución biológica (mutaciones y recombinaciones genéticas), así como también a una selección de acuerdo con algún criterio, en función del cual se decide cuáles son los individuos más adaptados, que sobreviven, y cuáles los menos aptos, que son descartados.

### Representación Binaria.

La representación usada por el algoritmo genético. La representación tradicional usada para codificar un conjunto de soluciones es el esquema binario en el cual un cromosoma<sup>1</sup> es una cadena de la forma (b1, b2, ..., bn), donde b1, b2, ..., bn 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 pionero de Holland en el área.

### Códigos de Gray.

La investigación en Algoritmos Genéticos fue que el uso de la representación binaria no mapea adecuadamente el espacio de búsqueda con el espacio de representación. La codificación de Gray es parte de una familia de representaciones. Podemos convertir cualquier número binario a un código de Gray haciendo XOR a sus bits consecutivos de derecha a izquierda. Por

### Codificación en números reales.

El espacio de búsqueda en el espacio de representación se vuelve más serio cuando tratamos de codificar números reales. Por ejemplo, si queremos codificar una variable que va de 0.35 a 1.40 usando una precisión de 2 decimales, necesitaríamos  $\log_2(140 - 35) \approx 7$  bits para representar cualquier número real dentro de ese rango.

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 real multiplicando y dividiendo cada dígito de acuerdo a su posición

## DESARROLLO

### Representación Binaria

```
C:\Users\Fernando\Desktop\7mo Semestre\GenÚticos\Pract2\C%4digo\Main.exe
Selecciona una opci%n:
1. Binario
2. Gray
3. Reales
4. Enteros
1
0 0 1 1 1
0 0 0 0 1
0 0 0 1 0
0 0 1 0 0
1 0 1 0 1
0 0 0 1 0
1 1 0 1 0
0 1 0 0 0
0 1 0 0 0
0 1 1 1 0

-----
Process exited after 21.04 seconds with return value 0
Presione una tecla para continuar . . .
```

### Representación en Gray

```
C:\Users\Fernando\Desktop\7mo Semestre\GenÚticos\Pract2\C%4digo\Main.exe
Selecciona una opci%n:
1. Binario
2. Gray
3. Reales
4. Enteros
2
0 1 0 1 0
0 1 0 1 0
0 1 0 0 0
1 1 1 0 0
1 1 1 0 1
1 1 0 0 0
0 0 1 0 0
1 0 1 0 1
1 1 1 0 0
0 0 0 0 0

-----
Process exited after 7.369 seconds with return value 0
Presione una tecla para continuar . . .
```

## Representación con Reales

```
C:\Users\Fernando\Desktop\7mo Semestre\GenÚticos\Pract2\C%4digo\Main.exe
Selecciona una opción:
1. Binario
2. Gray
3. Reales
4. Enteros
3
1.80 7.50 1.50 6.00 9.00
8.90 2.20 3.30 8.60 2.80
6.50 3.00 8.70 0.90 9.80
4.80 2.60 7.80 9.50 4.10
2.10 2.50 1.00 3.90 4.20
2.40 2.10 0.50 0.20 9.70
1.50 5.10 5.10 1.90 2.40
3.80 6.40 4.80 9.00 6.30
4.10 2.60 2.90 0.80 0.90
5.30 0.40 3.10 0.90 0.40

-----
Process exited after 56.81 seconds with return value 0
Presione una tecla para continuar . . .
```

## Representación con Enteros

```
C:\Users\Fernando\Desktop\7mo Semestre\GenÚticos\Pract2\C%4digo\Main.exe
Selecciona una opción:
1. Binario
2. Gray
3. Reales
4. Enteros
4
9 8 8 1 3
5 3 1 8 6
6 7 9 7 2
5 9 9 6 4
8 1 8 8 1
8 1 8 9 3
3 9 2 1 5
5 2 1 9 9
9 9 9 5 5
2 4 9 6 4

-----
Process exited after 8.871 seconds with return value 0
Presione una tecla para continuar . . .
```

## CONCLUSIONES

Fue una práctica sencilla, lo que más trabajo me costó fue hacer el código gray ya que lo convertí a base de números binarios, para que fuera realmente aleatorio.

Fue bastante fácil en general, el proceso de los 4 fue prácticamente lo mismo ya que se llena la matriz con números aleatorios y lo que varía es la generación de los mismos, por ejemplo, para el binario y el gray solo se generan números entre 0 y 1, para los reales genero dos veces números del 0 al 9, uno multiplicado por 0.1 y después los sumo, finalmente los enteros sólo genero números de 1 a 9.

C presenta muchas bondades ya que hay muchísimas bibliotecas de código abierto, y es fácil implementarlas, así podemos hacer muchas cosas, como en este caso el random.