



# Genetic Algorithms



## ***Practica N° 2 “Generación de una población de individuos con los distintos tipos de representación de soluciones potenciales en AG”***

### ***Objetivo:***

*Conocer y manejar los tipos de representación en algoritmos genéticos, comprender la importancia de éstos al implementar un algoritmo genético.*

### ***Fin:***

*Con el desarrollo de este programa, el estudiante deberá ser capaz de identificar, generar y operar la población de individuos, y proponer los tipos de representación de soluciones potenciales para resolver un problema.*

### ***Instrucciones.***

1. *Retoma el programa de la práctica #1 y diseña ahora el algoritmo para que pueda generar 4 tipos de representación de individuos, tal como se han explicado en clase. Puedes modificar el código de estos programas, es decir, basarte en la práctica #1 o bien puedes generar un nuevo programa en cualquiera de los lenguajes: C++, java, C# o Python.*
2. *Generación de una población de individuos:*
  - ✓ *Generar una población aleatoria de n individuos (mínimo 10 individuos, máximo 300), dependiendo del tipo de representación, estos deben ser de cualquiera de los distintos tipos de representación: Enteros, binarios, gray o reales.*
  - ✓ *Para cada tipo de representación, los números aleatorios a generar deben estar considerando en un rango especificado por el usuario, y para cada individuo y su cromosoma, cada alelo se crea de manera aleatoria. Veamos el ejemplo:*

#### **Representación con cadenas binarias**

- ✓ *En este caso, para encontrar el máximo de la función  $f(x) = 2x - 1.5$  evaluemos las representaciones de soluciones potenciales, en un AG, eligiendo entre el intervalo [0-15]*
- ✓ *Aquí representaremos a los individuos como "números binarios" de longitud 4, debido al máximo valor que contiene nuestro intervalo, ya que con un número binario de 4 bits podremos representar 16 números que se requieren: ( $2^4 = 16$ )*
- ✓ *Ejemplo:*
  - *Cromosoma para el individuo  $x = 5$ , tenemos: **00101***
  - *Cromosoma para el individuo  $x = 13$ , tenemos: **01101***

*Usar el mismo tipo de procesamiento respectivamente, para el resto de los tipos de representación.*

3. *Las buenas prácticas sugieren optar por generar un programa amigable para el usuario, por lo que al presentarle a este la generación de los tipos de representación, ésta debe ser cómoda, de fácil manejo, por ejemplo, con un menú.*
4. *El programa, además debe continuar evaluando la función  $f(x)$ , en este caso, solo para los tipos de representación binarios o gray (nótese que para evaluar a  $f(x)$ , tome su conversión entera).*
5. *Una vez evaluada la función, almacenar los resultados obtenidos y graficar los valores que maximizan y minimizan la función, ya que, de este proceso es de gran interés conocer que datos (números) hacen la función máxima.*
6. *Entrega de la evidencia:*
  - ✓ En el laboratorio: Una vez terminada la práctica, debes solicitar la revisión de la ejecución de la misma.
  - ✓ En la plataforma classroom: Elaborar un informe de la práctica y enviarla, el cual debe contener los siguientes datos:
  - ✓ Nombre al archivo como: **P2-CMME-FOR**, donde **P1** indica el número de práctica reportada, **MECM** corresponderá a iniciales de tu nombre (empezando por el apellido), y **.FOR** es el formato correspondiente al archivo, puede ser un formato *PDF, DOC, etc.*
  - ✓ Es importante agregar una portada como hoja de inicio que deberá incluir **Nombre de la unidad académica, Nombre de la unidad de aprendizaje, Número de la práctica, Objetivo de la práctica, Nombres de quien presenta, Nombre del(a) facilitador(A) (profesora) y Fecha de entrega.**
  - ✓ **Marco teórico:** breve información formal que sustenta el trabajo que se desarrolla (para este caso puede tomarla de la referencia que se anexa aquí (link) o consultar algún libro, si consulta alguna página, esta debe ser validada por alguna universidad u organismo educativo (paper o artículos, cursos en línea, presentaciones de docentes o investigadores, libros gratuitos en línea o por alguna editorial, etc.).
  - ✓ Desarrollo de la práctica en la cual debe anexarse el código del programa y una pantalla de la salida del mismo
  - ✓ Agregar las referencias consultadas para el desarrollo de la actividad.

## Antecedentes.

### “Tipos de representación usados en AG”

- **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 es una cadena de la forma  $(b_1, b_2, \dots, b_m)$ , donde  $b_1, b_2, \dots, b_m$  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.

1	0	0	1	1	0	1
---	---	---	---	---	---	---

- **Códigos de Gray.**

- La investigación en AGs 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 ejemplo, dado el número 101 en binario, haríamos 5:  $1 \oplus 0 = 1$ ,  $0 \oplus 1 = 1$ ,  $1 \oplus 0 = 1$ , produciéndose (el último bit de la izquierda permanece igual) 0111, el cual es el código de Gray equivalente. Algunos investigadores han demostrado empíricamente que el uso de códigos de Gray mejora el desempeño del AG.

- **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  $\log_2(140 - 35) \approx 7 \text{ bits}$  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.

2.15	1.89	0.43	3.14	0.27	7.93	5.11
------	------	------	------	------	------	------

- **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.

1	4	5	6	7	9
---	---	---	---	---	---