



UNIVERSIDAD NACIONAL DE INGENIERÍA

Facultad de Ciencias

Escuela Profesional de Ciencias de la Computación

Práctica dirigida: Fundamentos de los AE

Curso: Programación Evolutiva (CC521)

Semestre 2022-I

0.1. Representación de un decimal en binario

haciendo uso del [Google Colab](#) programe las siguientes funciones:

- Que convierta un número decimal a binario.
- Que pase un número binario a **código de gray**.
- Que pase un número en código de gray a binario

0.2. Generar individuos

Usando las funciones anteriores, elaborar el pseudocódigo e implementar el código para generar "individuos/cadenas binarias", de los siguientes dominios.

- Para todo valor entero de x en el rango de $[-5, 5]$, generar las cadenas binarias y en código de gray.
- Para valores discretos de x con una precisión de 10^{-3} en el rango de $[-5, 5]$, generar 10 cadenas binarias y en código de gray de forma aleatoria.
- Para valores discretos de x e y con una precisión de 10^{-3} donde $x \in [-5, 5]$ e $y \in [0, 10]$, generar 10 cadenas binarias y en código de gray de forma aleatoria.

0.3. Construyendo los operadores básicos

Usando las funciones anteriores, implementar el código para los siguientes casos:

- Función población inicial, los individuos se deben elegir de forma aleatoria.
- Función de selección, la selección lo debe realizar de forma aleatoria.
- Función de Recombinación, debe elegir un punto aleatorio dentro de la cadena y hacer el cruce para generar los descendientes.
- Función de mutación, debe elegir un bit o una posición de la cadena y invertirlo el valor, es decir, si tiene cero cambiarlo a 1 o viceversa.
- Por último, implementarlo el algoritmo de canónico.

0.4. El problema *OneMax* (o *One-Max*) es una tarea de optimización simple que se utiliza a menudo como el ejemplo Hola mundo en los frameworks de algoritmos genéticos.

La tarea de **OneMax** es encontrar la cadena binaria de una longitud determinada que maximice la suma de sus dígitos. Por ejemplo, el problema **OneMax** de longitud 5 considerará candidatos como los siguientes:

- 10010 (suma de dígitos = 2)
- 01110 (suma de dígitos = 3)

- 11111 (suma de dígitos = 5)

Obviamente (para nosotros), la solución a este problema es siempre la cadena que comprende todos los 1. Pero el algoritmo genético no tiene este conocimiento y necesita buscar ciegamente esta solución utilizando sus operadores genéticos. Si el algoritmo hace su trabajo, encontrará esta solución, o al menos una cercana, dentro de un período de tiempo razonable.

0.5. Optimización (maximización) de una función $f(x)$

Consideremos la función $f(x) = x\sin(x) + x\cos(x)$ donde x pertenece al dominio de $[0, 5]$.

- Diseñar un algoritmo evolutivo iterativo (generaciones) para el encontrar el valor óptimo (valor máximo) con una precisión de 10^{-3} .
- Elabore un programa en python siguiendo la secuencia del algoritmo diseñado en el ítem anterior.

0.6. Optimización (maximización) de una función $f(x)$

Para el problema anterior implementar la codificación del cromosoma como código de Gray, elaborar el pseudocódigo e implementar el código en un cuaderno de notas