

# Universidad de Carabobo Facultad Experimental de Ciencias y Tecnología Departamento de Computación



Electiva: Introducción a la Inteligencia Artificial

## Tarea 1: Algoritmos Genéticos simples

Integrantes:

Jhonattan Garcia, C.I: 24.423.188

Luis Sierra, C.I: 25.582.661

Prof. Aníbal Guerra

Bárbula, 07 de septiembre de 2021

#### Pasos para ejecutar el programa

1- Tener instalado el intérprete de Python en su S.O.

2- Crear un entorno virtual con el siguiente comando:

Windows: py -m venv venvLinux: python3 venv venv

3- Ejecutar el programa

Windows: py main.pyLinux: python3 main.py

#### Datos utilizados para el programa

Los datos de prueba se encuentran un el archivo **in.txt** donde cada línea corresponde a un caso de prueba distinto. Cada línea contiene 5 elementos separados por un espacio en blanco que corresponden a: N, Pc, Pm, #iter, Gap.

#### Entrada:

N	PC	PM	#iter	Gap
10	0.1	0.001	100	10
100	8.0	0.5	1000	100
10	0.1	0.5	100	100
10	8.0	0.001	100	100
10	8.0	0.001	100	10
100	0.1	0.5	1000	100
100	0.1	0.5	1000	10
100	8.0	0.001	1000	100
100	8.0	0.001	1000	10

<u>Salida:</u> para casa de prueba se generará un archivo outN.txt con N = 1, 2..., n correspondiente al reporte de las generaciones.

#### Observaciones

#### <u>Generales</u>

- 1. El programa puede tardar en generar el archivo de salida ya que al realizar un proceso de cruce o mutación se pueden crear individuos que no pertenezcan al espacio de soluciones y de ser así, se repetirá dicho proceso hasta que aparezcan los individuos permitidos dentro del rango [0, 1].
- 2. No todas las veces que se ejecuten los mismos casos de prueba se obtienen los resultados esperados.

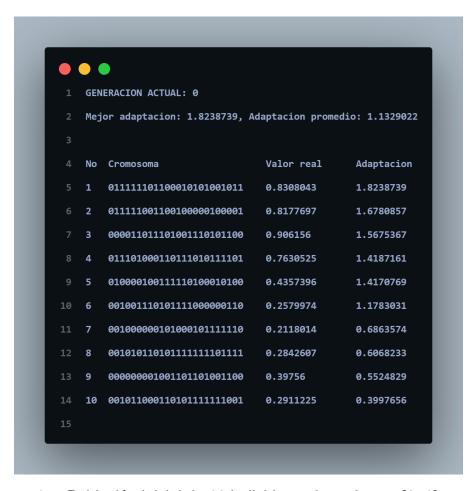
#### Casos de prueba adicionales

1.

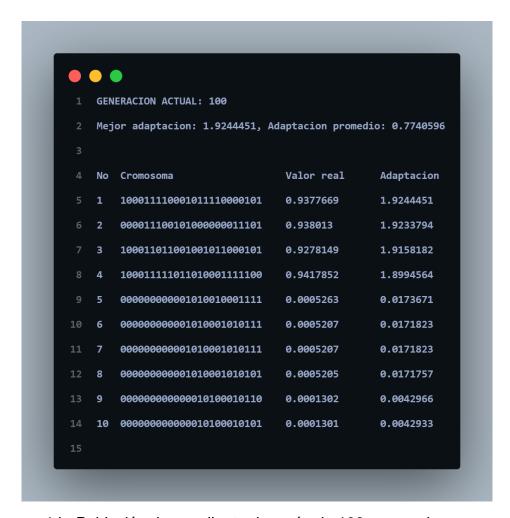
N	PC	PM	#iter	Gap
10	0.1	0.006	100	60

Al realizar el experimento se comprobó que utilizando valores mayores a 60% para el Gap y un número mayor a 100 generaciones, el tiempo de respuesta del programa es mucho más lento debido a la cantidad de individuos nuevos que van apareciendo a través de las nuevas generaciones. Adicionalmente, se verifico con una cantidad menor a 100 generaciones, pero se observó que el resultado no es el esperado.

Las imágenes que se presentan a continuación permiten visualizar la convergencia de la función con los datos introducidos.



1.a. Población inicial de 10 individuos aleatorios en [0, 1].



1.b. Población descendiente después de 100 generaciones.

2.

N	PC	PM	#iter	Gap
100	0.7	0.002	100	20

Con este experimento observamos que a medida que se incrementa la cantidad de individuos, si la cantidad de generaciones es mayor a 100 y el Gap es mayor al 20%, el tiempo de respuesta es lento.

En las imágenes que se muestran a continuación se puede apreciar la convergencia de la función con estos parámetros.

```
GENERACION ACTUAL: 0
    Mejor adaptacion: 1.9258245, Adaptacion promedio: 1.1394185
                                    Valor real
    No Cromosoma
                                                    Adaptacion
        000011100011000000111110
                                    0.929854
                                                    1.9258245
        100011011010101001001111
                                    0.9284175
                                                    1.9192149
        100011111000101011001110
                                    0.9407182
                                                    1.9076581
        011111110001000010001001
                                    0.8327305
                                                    1.8311539
        1000101111001000000000101
                                    0.9160709
                                                    1.7784422
        100000010011110000110011
   6
                                    0.8469555
                                                    1.7683905
        100000010110111111010110
                                    0.8482774
                                                    1.7524578
   8
        100010101110001010010101
                                    0.9101973
                                                    1.6627828
        011011101001100111101010
                                    0.7248362
                                                    1.6581773
    10 100000101001100010100101
                                    0.8558757
                                                    1.6306206
    11 011000001101011110011100
                                    0.6346652
                                                    1.6285037
    92 000111111110010100101001
                                    0.2090281
                                                    0.6037029
    93 0000110101000100000000101
                                    0.0869381
                                                    0.4388131
    94 000011010111000010001001
                                    0.0880777
                                                    0.4055913
   95 000001000111010101000101
                                    0.292165
                                                    0.3675913
    96 000011111101110101101101
                                    0.1039725
                                                    0.2884374
    97 000011101000100001001011
                                    0.0952395
                                                    0.1890301
    98 000011101011101011011101
                                    0.0965341
                                                    0.1490114
    99 000011101110011110010000
                                    0.0976784
                                                    0.1135616
    100 000011101110111100000100
                                    0.0978692
                                                    0.1076473
```

2.a. Población inicial de 100 individuos aleatorios en [0, 1].

```
GENERACION ACTUAL: 100
    Mejor adaptacion: 1.9258245, Adaptacion promedio: 1.6186474
                                    Valor real
    No Cromosoma
                                                    Adaptacion
        000011100011000000111110
                                    0.929854
                                                    1.9258245
        100011011010101001001111
                                    0.9284175
                                                    1.9192149
    2
        100011111000101011001110
                                    0.9407182
                                                    1.9076581
        011111110001000010001001
                                    0.8327305
                                                    1.8311539
        000011001101101011011000
                                    0.842456
                                                    1.8101056
        000011001110011101011001
   6
                                    0.845657
                                                    1.7824363
        1000101111001000000000101
                                    0.9160709
                                                    1.7784422
        100000010011110000110011
                                    0.8469555
                                                    1.7683905
        100000010110111111010110
                                    0.8482774
                                                    1.7524578
    10 011100000101010010101000
                                    0.7361704
                                                    1.7361603
    11 0111000010111101010101111
                                    0.7388503
                                                    1.7355501
    92 011001100101001110110001
                                    0.6706097
                                                    1.1775558
    93 000010100011110000000111
                                    0.670727
                                                    1.174434
    94 001101111000101011000101
                                    0.3640005
                                                    1.1586042
   95 001110001101110000110101
                                                    0.9713779
                                    0.3726389
    96 000001011010111110110010
                                    0.372658
                                                    0.9709074
    97 011001111110110011000011
                                    0.6810819
                                                    0.876347
    98 011001111110111111101000
                                    0.6811624
                                                    0.8739005
    99 01001100101001011100100
                                                    0.8600787
                                    0.5023076
    100 001011000110011101010010
                                    0.2910034
                                                    0.4034343
```

2.b. Población descendiente después de 100 generaciones.

### Listado del programa

Función	Entrada	Salida
float_to_bin	<ul><li>x: número real,</li><li>n_bits: cantidad de bits</li><li>permitidos</li></ul>	Retorna el equivalente en binario de x
créate_population	<ul><li>n: cantidad de individuos,</li><li>n_bits: cantidad de bits</li><li>permitidos</li></ul>	Genera una población de N individuos únicos.
evaluate	population: lista de individuos	Agrega el valor de la adaptación a cada individuo de la población.
add_adaptations	population: lista de individuos	Retorna la suma de todas las adaptaciones de los individuos que forman la población de entrada
show_progress	population: lista de individuos, cases: número de caso de prueba actual, generations: generación actual	Muestra la estadística de la generación actual

selection_probability	population: lista de	Agrega el valor de la
	individuos	probabilidad de selección
		a cada individuo de la
		población
search	search_space: espacio	Retorna el índice del
	de búsqueda que	individuo donde se
	corresponde a la lista de	encuentre elem
	la rueda de la ruleta,	
	elem: elemento a	
	comparar con los valores	
	del espacio de búsqueda	
roulette_wheel	population: lista de	Retorna dos individuos
	individuos	aleatorios
encode	num: número real,	Dado un número real,
	n_bits: umero de bits	obtiene su
	permitidos	correspondiente número
		entero
decoding	num: número entero,	Dado un número entero,
	n_bits: umero de bits	obtiene su
	permitidos	correspondiente número
		real

scientific_notation	num: número real	Convierte el número real
		de formato x.xxe-0x a
		0.0000xxx
modification_point	prob: probabilidad de	Retorna el punto de
	cruce o mutación,	modificación de bits para
	n_bits: cantidad de bits	el cruce y la mutación
	permitidos	
crossing	parent1 y parent2:	Retorna 2 hijos de dos
	números reales que	individuos
	representan a los padres,	
	pc: probabilidad de	
	cruce,	
	n_bits: cantidad de bits	
	permitidos	
mutation	p1 y p2: números reales	Retorna la mutación de
	que representan a los	p1 y p2
	individuos,	
	pc: probabilidad de	
	cruce,	
	n_bits: cantidad de bits	
	permitidos	
main		Cuerpo principal del AGS