



1. Objetivo del Laboratorio

*Desarrollar de forma autónoma un algoritmo genético que permita optimizar una función o problema a partir de una población de individuos generadas aleatoriamente a los que se le aplica un mecanismo evolutivo (**selección** + **operador genético**) definido por unos parámetros.*

Debes de usar comentarios con profusión, incluyendo celdas específicas donde expliques los algoritmos que usas y el código que has programado. Cuantos más comentarios haya, mejor será la evaluación y, probablemente, menos preguntas serán necesarias en la defensa de la práctica.

2. Elementos a utilizar:

- Lenguaje Python
- Librerías: numérica *NumPy*, estructuras de datos *pandas*, gráfica *Matplotlib* (opcional si se quieren implementar gráficas).
- Entorno Anaconda
- Editor Jupyter

3. Práctica 1 (AG frases en castellano con mutaciones)

Objetivo

Construye un algoritmo genético genérico que sirva para optimizar problemas en los que el objetivo sea conseguir una determinada frase objetivo (o target) usando únicamente el mecanismo de mutaciones. El código seguirá las instrucciones del apartado “**Implementación**” de esta práctica. Responde después a las preguntas que se plantean en “**Cuestiones**”.

Implementación

Crea el notebook *LAP1-Frase.ipynb*. Ten en cuenta lo siguiente:

- Un cromosoma codificará una frase en castellano, donde cada letra de la frase será un gen.
- Cada gen podrá valer cualquiera de las letras del alfabeto (solo MAYUSCULAS) + espacio.
- Los únicos datos que has de suministrar al programa son:
 - o **Sentencia target:** de longitud arbitraria
 - o **NPOB:** Número de individuos de la población
 - o **NGEN:** Número de generaciones (ciclo completo sobre todos los individuos) de la prueba
 - o **P_m:** Probabilidad de Mutación (probabilidad de que un gen dado mute)
 - o **NRES:** Cada qué número de generaciones se saca un resumen de la evolución del proceso



- o **NSAMPLE**: Cada qué número de generaciones se saca un muestreo de la población

El programa en Python deberá responder a los siguientes puntos:

1. Generar una población de individuos aleatorios y calcular su **fitness**, que depende del **número de coincidencias** (caracteres correctamente colocados) **entre él y la sentencia target** según la fórmula

$$F_i = e^{(ncoin_i - ltar)} - e^{(-ltar)}$$

Donde

- $ncoin_i$ es el número de coincidencias con la Target del individuo i -ésimo
- $ltar$ es la longitud de la Target (**hay que calcularla**)

2. En cada generación, secuencialmente sobre toda la población y comenzando por un individuo elegido **al azar**,
 - a. Decidir si el *cromosoma padre* se replicará o no usando un Método de Montecarlo simple sobre el valor de **Probabilidad de Selección** (P_s) obtenido como sigue:

$$P_{s_i} = \frac{F_i}{e^{(ncoin_{max} - ltar)} - e^{(-ltar)}}$$

Donde $ncoin_{max}$ es el número de coincidencias de la frase que más se parece a la Target

- b. Elegir al azar otro individuo de la población, el cual será borrado y sustituido por el individuo recién creado (¡puede ser él mismo!)
 - c. Secuencialmente para cada uno de los genes del *padre* decidir si cada gen se copia fielmente en el hijo o se muta (cambia) usando un Método de Montecarlo simple sobre el valor de P_m . Si se muta, se elige otro carácter al azar escogido entre el conjunto de caracteres permitidos (¡puede volver a ser el mismo!)
 - d. Calcular el *fitness* del nuevo individuo
3. Escribir la siguiente información:
 - o Cabecera con los parámetros usados: **Sentencia Target**, **ltar**, **NPOB**, **NGEN**, P_m
 - o Cada NRES generaciones escribir un resumen que contenga:
 - N° de generación
 - El **mejor individuo**, su número de coincidencias $ncoin_{max}$ y cuantos hay en número (**NTar**) y en % (**%Ntar**)
 - **Individuo consenso** (aquel obtenido con el carácter más frecuente en cada posición) y el valor de $ncoin_{media}$
 - o Cada NSAMPLE generaciones escribir
 - El resumen NRES
 - Un muestreo del 20% de los individuos de la población

Cuestiones

Elabora una memoria de la práctica en la que respondas a las siguientes cuestiones

1. Explica detalladamente el código que implementa las funciones de cálculo de F_i , P_s , elección de individuo a borrar, replicación/mutación de un individuo.



2. Estudiar qué pasa con $\%NTar$, $NTar$, $ncoin_{max}$ y $ncoin_{medio}$ respecto a $ltar$, $NPOB$ y $NGEN$ (probando frases de longitud entre 5 y 25 caracteres). Utilizar gráficos donde se recoja la relación entre estas variables e intentar encontrar alguna relación que garantice para un $ltar$ concreto el mejor resultado. Explica lo que ocurre.
3. Para la frase *ANIMULA VAGULA BLANDULA*, estudia que ocurre con $\%NTar$, $NTar$, $ncoin_{max}$ y $ncoin_{medio}$ para valores crecientes de P_m . Explica la gráfica que obtienes. Estudia también qué pasa con el **individuo consenso**.

4. Práctica 2 (AG frases en castellano con mutaciones y elitismo)

Objetivo

Modifica el algoritmo genético anterior para incluir elitismo siguiendo las instrucciones del apartado “**Implementación**” de esta práctica. Responde después a las preguntas que se plantean en “**Cuestiones**”.

Implementación

Crea el notebook *LAP2-Frase_Elite.ipynb*., copia el programa de la práctica anterior y modifícalo de forma que la selección de los individuos que se van a replicar en cada generación **se haga únicamente escogiendo el % mejor de la población** y de cada uno de ellos generando 4 hijos para obtener de nuevo la población completa que se usará en la siguiente generación.

En cada generación,

1. Escoge un **% de los mejores individuos de la población** (aquellos con el valor de Fitness más alto)
2. De cada uno de ellos genera el número suficiente de hijos para obtener de nuevo la población completa que se usará en la siguiente generación (los padres no pasan tal cual a la siguiente generación).
3. Los hijos de cada individuo seleccionado se generan decidiendo si cada gen se copia fielmente en el hijo o se muta usando un Método de Montecarlo simple sobre el valor de P_m (igual que en la práctica 1)

Cuestiones

Continúa en la memoria de la práctica respondiendo a las siguientes cuestiones

1. Prueba con distintos porcentajes de **elitismo** y encuentra cual es el mejor. Haz una tabla comparativa y **explica** detalladamente el código implementado y los detalles de la configuración elegida.
2. Responde a la cuestión 2 y 3 de la práctica anterior con los resultados obtenidos con el % de elitismo seleccionado en la pregunta anterior.
3. Compara los resultados obtenidos en ambas prácticas entre sí. Encuentra explicación al distinto comportamiento (si lo hubiera) y extrae las correspondientes conclusiones.

5. Forma de entrega del laboratorio:

La entrega consistirá en un fichero comprimido RAR con nombre **LAB04-GRUPOxx.RAR** subido a la tarea **LAB4** que **contenga únicamente**



1. **Por cada práctica** un notebook de Jupyter (archivos con extensión **.ipynb**).
2. Una **memoria del laboratorio** en Word.

Las entregas que no se ajusten exactamente a esta norma NO SERÁN EVALUADAS.

6. Rúbrica de la Práctica:

1. IMPLEMENTACIÓN: Multiplica la nota del trabajo por 0/1

Siendo una práctica de IA, todos los aspectos de programación se dan por supuesto. La implementación será:

- Original: Código fuente no copiado de internet. Grupos con igual código fuente serán suspendidos
- Correcta: Los algoritmos están correctamente programados. El programa funciona y ejecuta correctamente todo lo planteado en el apartado “*Cuestiones*” de cada práctica.
- Comentada: Inclusión (**obligatoria**) de comentarios.

2. MEMORIA DEL LABORATORIO

Obligatorio redacción clara y correcta ortográfica/gramaticalmente con la siguiente estructura:

- *Portada con el nombre de los componentes del grupo y el número del grupo*
- *Índice*
- *Resultados de la Práctica 1*
- *Resultados de la Práctica 2*
- *Bibliografía*

Calificación de las cuestiones:

PRÁCTICA	CUESTIÓN	VALORACIÓN (sobre 10)
Práctica 1	Cuestión 1	1,5
	Cuestión 2	1
	Cuestión 3	2
Práctica 2	Cuestión 1	1,5
	Cuestión 2	1
	Cuestión 3	2