UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS

Proyecto Final

para la materia "Búsqueda de Soluciones e Inferencia Bayesiana"

Encriptación evolutiva usando el Cifrado de Vigenère

Asesor: Dr. Jorge Hermosillo Valadez

Estudiante: Giulia López Maldonado

Licenciatura en Inteligencia Artificial

Centro de Investigación en Ciencias
Universidad Autónoma del Estado de Morelos
Av. Universidad 1001
Col. Chamilpa, C.P. 62209
Cuernavaca, Morelos, México
giulia.lopez@uaem.edu.mx

Contents

Encriptación evolutiva usando el Cifrado de Vigenère

August 9, 2025

Abstract

El cifrado desarrollado por Blaise de Vigenère, durante el siglo XVI, fue considerado indescifrable durante muchos años, lo que dio orígen a su nombre 'el cifrado indescifrable'. Y a pesar de su vulnerabilidad, en la actualidad el cifrado de, Vigenère tuvo un impacto significativo en la criptografía.

En este trabajo, se presenta un algoritmo genético en Python diseñado para optimizar la clave de encriptación de un mensaje específico utilizando el cifrado de, Vigenère. Los resultados muestran que el uso de algoritmos genéticos puede mejorar la seguridad y eficiencia del cifrado de Vigenère, al optimizar la selección de claves, tomando en cuenta distintos criterios para evaluar, qué hace que una clave cifre un texto plano mejor que otra.

1 Introducción

En el ámbito de la criptografía, el cifrado Vigenère se destaca por su simplicidad y, durante mucho tiempo, su aparente indescifrabilidad. Desarrollado en el siglo XVI por Blaise de Vigenère, este método de cifrado polialfabético utiliza una tabla de caracteres para transformar un texto plano en un texto cifrado. A pesar de su robustez inicial, con el tiempo se han desarrollado métodos para romperlo, como el análisis de frecuencias. En este proyecto, se explora la optimización de claves de cifrado utilizando algoritmos genéticos, una técnica que imita el proceso de selección natural para encontrar soluciones óptimas.

1.1 El cifrado de Vigenère

El cifrado de Vigenère es un cifrado desarrollado en el siglo XVI por el criptógrafo francés Blaise de Vigenère. Su nombre se debe a su contribución al desarrollo y popularización del cifrado. Según refiere la Universidad de Granda "*El cifrado Vigenère es un método poli alfabético de sustitución, basado en una tabla de caracteres conocida como tabla de Vigenère, derivada de la tabula recta de Trithemius.*"[3].

El cifrado de Vigenère fue muy aclamado y se consideró indescifrable durante muchos años, lo que le valió el nombre de 'le chiffre indéchiffrable' o 'el cifrado indescifrable' en francés. Sin embargo, a mediados del siglo XIX, el matemático Charles Babbage descubrió un método para descifrarlo utilizando el análisis de frecuencias.

Blaise de Vigenère mejoró el anterior cifrado inventado por León Battista Alberti, en el siglo XV. El cifrado Alberti, utilizaba un alfabeto mixto único y un dispositivo de 'disco cifrado' para cifrar y descifrar. La mejora de Vigenère, introdujo el uso de una palabra clave o frase repetida para determinar los turnos, convirtiéndolo en un cifrado polialfabético más seguro en comparación con los cifrados monoalfabéticos como el cifrado César.

Este cifrado funciona utilizando una palabra clave como base para el cifrado, misma que se repite para que coincida con la longitud del mensaje en texto plano. Cada letra de la palabra clave se utiliza para determinar el valor de desplazamiento de la letra correspondiente en el texto sin cifrar.

Para cifrar un mensaje, cada letra del texto plano se desplaza por la letra correspondiente de la palabra clave. Para ello se utiliza una tabla llamada cuadrado de Vigenère(Figura 1), una cuadrícula de alfabetos en la que cada fila representa un desplazamiento de la fila anterior en una posición. La letra en la intersección de la letra de la palabra clave y la letra del texto plano en el cuadrado de Vigenère da la letra cifrada.

1.1.1 Ejemplo de cifrado Vigenère

A continuación se muestra un ejemplo que ilustra cómo funciona el cifrado Vigenère.

Supongamos que, deseamos cifrar la palabra "MESA" con la clave de encriptación "DIO". El cifrado Vigenère opera de la siguiente manera:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	D
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	C
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	B
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Figure 1: Tabla de Vigenère.

1. **Asignación de clave repetida**: La clave "DIO" se repite tantas veces como sea necesario para cifrar todo el mensaje. En este caso, se repetirá para cubrir las cuatro letras de "MESA".

2. **Proceso de cifrado por letra:** La primera letra del mensaje "M" se cifra usando la primera letra de la clave "D". La intersección correspondiente en la tabla de Vigenère da como resultado "P". El proceso se repite con cada una de las letras cómo puede observarse en la Figura 2.

3. F**ormación del mensaje cifrado**: Combinando estos resultados, el mensaje cifrado para "MESA" con la clave "DIO" sería "PMGD".

1.2 Formalización del Cifrado de Vigenère.

Anteriormente, se explicó el funcionamiento para entender humanamente este cifrado, sin embargo, es oportuno dar una definición matemáticamente formal para el planteamiento del problema. Existen dos fórmulas propuestas por Vigenère, para el cifrado y decifrado de su método. Dichos procedimientos están dados por:

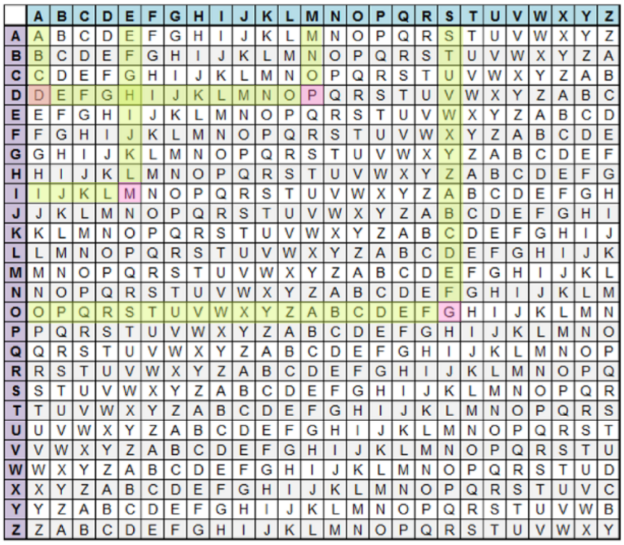

Figure 2: Ejemplo de cifrado Vigenère

Cifrado:

$$C[i] = (P[i] + K[i]) mod 26$$

Descifrado:

$$P[i] = (C[i] - K[i]) mod 26$$

Dónde:

- $C[i]$ es el i-ésimo carácter del texto cifrado.
- $P[i]$ es el i-ésimo carácter del texto plano convertido a su posición en el alfabeto (A=0, B=1, ..., Z=25).
- $K[i]$ es el i-ésimo carácter de la clave convertido a su posición en el alfabeto (A=0, B=1, ..., Z=25).

1.3 Problemática

A simple vista, este cifrado parece ser uno intuitivo y fácil de descifrar, sin embargo, sin contar con la clave para desencriptar un mensaje, parece *casi* imposible descifrar este mismo. Las **claves de encriptación**, tienen **efectos distintos** para cada texto a cifrar correspondientemente. Es decir, una clave

de cifrado de longitud 10, con la misma letra repetida varias veces, no tiene el mismo impacto en la encriptación de un mensaje de longitud 10, que una clave de longitud 3 pero con letras distintas. Uno podría pensar que, una clave de longitud mayor o igual a la de la longitud del texto plano, resulta encriptarlo de mejor manera que una clave de longitud menor a la del texto plano, más no resulta así. Es pertinente mencionar que, la mayor parte de los métodos para desencriptar Vigenère, se enfocan mayormente en la cantidad de repeticiones y patrones en un texto, que en la longitud de la clave en sí, lo que varía según las coincidencias entre el texto plano y el cifrado, así como la repetición de una o varias letras en la clave de encriptación.

Es por esto, que, una clave de encriptación no puede definirse según un listado de reglas, o a simple intuición humana. Un algoritmo evolutivo resulta útil para la definición de claves para un texto plano en específico. De esta manera, desencriptar un texto sin conocer la clave de encriptación, parece casi imposible con los métodos más famosos que rompen este cifrado. Siendo qué, para cada texto existirá una clave óptima que lo encripte de mejor manera de entre una población aleatoriamente generada.

2 Objetivos

Cómo se mencionó con antelación, el objetivo general es, realizar un algoritmo evolutivo en Python, que encuentre la clave óptima para cifrar un mensaje en específico, utilizando el cifrado de, Vigenère.

Ahora que se ha definido el objetivo general de este proyecto, resulta conveniente enumerar los objetivos específicos que sirven como un enfoque de lo que desea lograrse:

1. Elegir los métodos de reproducción, selección y mutación adecuados que se adapten y solucionen el problema de mejor manera.

2. Probar distintas funciones que midan la aptitud de cada individuo en la población de cada generación (que es una posible clave para cifrar un mensaje específico) y encontrar la que modele de mejor forma la problemática propuesta.

3. Analizar patrones en las longitudes y características de las distintas claves elegidas durante distintas compilaciones del código en busca de patrones para poder inferir sobre el comportamiento de este.

3 Metodología

3.1 Representación de los individuos

Cada individuo de la población estará representado como una cadena de caracteres (letras del alfabeto) de distintas longitudes. Asimismo, cada individuo tendrá asociado el texto cifrado con la clave del individuo.

Como la longitud de las cadenas juega un papel importante, resultaría ideal permitir que, las cadenas tuvieran longitudes extensas, sin embargo, por la complejidad computacional que esto representaría, se tomará en cuenta un rango, dado por la longitud del texto plano mismo, para evitar tener generaciones de cadenas que complicarían y alentarían el funcionamiento del algoritmo. De esta forma, la longitud máxima de cada cadena será la tercera parte de la longitud el texto plano.

3.2 Función de Aptitud

Definir una medida de rendimiento para cada individuo representa un desafío significativo, dado que no existe una norma o estándar universal para evaluar una clave de encriptación. ¿Cómo puede distinguirse entre una clave que encripta un texto plano de manera segura y óptima? Al abordar esta cuestión, se expone que, a lo largo del desarrollo de este proyecto, se han considerado diversas funciones para medir la aptitud de las claves. No obstante, se dará prioridad a dos métodos principales:

3.2.1 Método de Kasiski

Durante aproximadamente 300 años se creyó que el cifrado de Vigenère era irrompible, aunque Charles Babbage y Friedrich Kasiski determinaron de forma independiente un método para romperla a mediados del siglo XIX. Este método utiliza patrones repetidos en el texto para determinar la longitud de la clave. Una vez que se sabe que, la clave tiene "n" caracteres, se puede aplicar el análisis de frecuencia a cada "n" letras para determinar el texto en claro. A pesar de que el método para descifrar el cifrado de Vigenére se había publicado 50 años antes, hasta la década de 1920, en un artículo publicado en Scientific American consideró que, "*El cifrado era indescifrable y se describió como 'imposible de traducir*" (Scientific American, 1917)

Los pasos que sigue este método según Arboledas, D. en su libro "Criptografía sin secretos con Python"[1] son:

1. Encontrar todos los n-gramas repetitivos dentro del texto codificado.
2. Calcular la distancia entre los criptogramas repetitivos.
3. Calcular todos los factores (divisores) de la distancia (el factor divisor expresa la longitud de la clave).
4. Determinar las divisiones del conjunto de factores divisores. El valor que aparece en la división representa el número que aparece en todos los factores divisores de las distancias. Este valor puede ser la longitud de la clave. Esto se debe a que las cadenas repetidas pueden aparecer superpuestas.

Ahora que se tiene un mejor entendimiento del funcionamiento de este método, es que puede plantearse su utilización como función de aptitud. El método de Kasiski calcula un aproximado de la longitud de la clave con la que se cifró el texto tomado como entrada.

La aplicación de este método como función de aptitud consiste en calcular el valor absoluto de la diferencia entre la longitud real de la clave del individuo y la longitud estimada por el método Kasiski. Un valor de aptitud mayor indica que el método Kasiski aproximó incorrectamente la longitud de la clave, sugiriendo que la clave es más segura y difícil de romper. Sin embargo, el método de Kasiski a menudo falla, debido a la superposición de cadenas repetidas, lo que afecta negativamente su precisión y utilidad.

3.2.2 Conteo de repeticiones de una letra, según un determinado número de desplazamientos

Dada la limitación del método de Kasiski, se plantea una función de aptitud alterna. Esta nueva función busca minimizar y no maximizar el valor de aptitud, proporcionando una evaluación más robusta de la seguridad de la clave.

El proceso consiste en contar las repeticiones de una letra específica en el texto cifrado después de aplicar diferentes desplazamientos. Si el texto cifrado presenta muchas repeticiones de la misma letra en ciertos desplazamientos, esto sugiere que la clave utilizada es menos segura, ya que el texto cifrado contiene patrones fácilmente identificables. Por el contrario, una clave más

segura generará un texto cifrado con menor repetición de letras en cualquier desplazamiento, lo que dificulta su análisis.

En este método, se toman como entrada el texto plano y el número de desplazamientos a realizar, y los pasos a seguir son:

1. Se realizan el número de desplazamientos requeridos, y se guardan las distintas combinaciones en una matriz, es decir, si se da como entrada la cadena "AATOPAOTP", se calcularía la matriz de desplazamientos, con 7 desplazamientos:

Table 1: Matriz de desplazamientos

A	A	T	O	P	A	O	T	P
A	T	O	P	A	O	T	P	A
T	O	P	A	O	T	P	A	A
O	P	A	O	T	P	A	A	T
P	A	O	T	P	A	A	T	O
A	O	T	P	A	A	T	O	P
O	T	P	A	A	T	O	P	A

2. Utilizando la matriz de desplazamientos, se calcula la frecuencia máxima de letras en cada columna. Esto implica contar cuántas veces aparece la letra más común en cada columna de la matriz. Este paso es crucial para detectar repeticiones que podrían indicar la longitud de la clave en un cifrado polialfabético. Por ejemplo, en nuestro caso, la mayor incidencia de una letra en cada columna de la matriz de desplazamientos sería: $[3, 2, 2, 2, 3, 3, 2, 2, 3]$

3. Finalmente, se toma el máximo valor de la lista obtenida en el paso anterior. Este valor representa la cantidad máxima de incidencias de la letra más frecuente en cada columna.

De esta manera, cada individuo tendría asociado un valor de aptitud dado por la cantidad máxima de incidencias de la letra más frecuente en la matriz de desplazamientos. Esta función de aptitud se plantea hacer lo contrario al método de Kasiski: los individuos con el valor mínimo de incidencias máximas representan claves de cifrado más óptimas y seguras en comparación con aquellos que tienen valores altos.

4 Marco teórico

La elección de un algoritmo evolutivo para la búsqueda de la clave óptima para encriptar un mensaje en específico, tiene como fundamento muchos aspectos tomados en cuenta, principalmente, y como se profundizó en el punto anterior, es realmente complicado determinar qué hace que un individuo sea *mejor* para encriptar un mensaje que otro, por lo que no puede elegirse a simple vista de entre una población de individuos con longitudes y contenidos arbitrarios. Por otro lado, es vital la aleatorización en el proceso de búsqueda, desde la creación hasta el momento de elegir los mejores individuos, debido a que, al ser claves tan diversas en varios aspectos, es posible que el algoritmo converja a soluciones tempranas por falta de diversidad.

Algoritmo genético

Los Algoritmos Genéticos (AGs) son métodos adaptativos utilizados para resolver problemas de búsqueda y optimización, basados en el proceso genético de los organismos vivos. Inspirados en los principios de la selección natural y la supervivencia del más apto postulados por Darwin en 1859, los AGs imitan estos procesos naturales para crear soluciones a problemas del mundo real.

A lo largo de generaciones, las poblaciones en la naturaleza evolucionan adaptándose a su entorno. De manera similar, los AGs evolucionan las soluciones hacia valores óptimos mediante una adecuada codificación y manipulación de estas. La metáfora subyacente en los AGs es la evolución natural, donde cada especie se modifica en la búsqueda de adaptaciones beneficiosas a un ambiente complejo y cambiante. El conocimiento de cada especie está incorporado en la estructura cromosómica de sus miembros.

Por esta razón, los AGs adoptan la terminología de la genética natural y se utilizan principalmente en problemas de optimización y búsqueda de máximos (o mínimos), aunque no están limitados exclusivamente a esta clase de problemas.

Para aplicar un algoritmo genético a un problema específico, se deben considerar los siguientes cinco componentes esenciales:

- **Representación de los individuos**. Es fundamental definir una forma efectiva de representar la población inicial de soluciones potenciales. Las representaciones comunes incluyen binarias, enteras, reales, alfabéticas y permutacionales. Sin embargo, es útil modelar los individuos según las características y necesidades de cada problema específico.

- **Función de aptitud**. Esta función evalúa la aptitud de los individuos, determinando qué tan buena es una solución en comparación con otras. La función de aptitud debe estar bien diseñada para reflejar con precisión la calidad de las soluciones.

- **Función de selección**. Una función de selección actúa como el entorno, clasificando a los individuos según su adaptación y aptitud. Entre los métodos más comunes se encuentran la selección elitista, la rueda de ruleta, los torneos y la selección por rangos. Es importante conocer el tipo de población que maneja cada problema para elegir la selección que modele adecuadamente a los individuos.

- **Función de reproducción**. Esta función define un método para cruzar los mejores individuos y mantener sus características útiles en la siguiente generación, garantizando la propagación de buenas soluciones. Las formas comunes de cruzamiento incluyen el cruzamiento de un punto, de dos puntos y el cruzamiento uniforme.

- **Operadores genéticos**. Los operadores genéticos alteran la composición de los descendientes. La estrategia más sencilla es la mutación, que cambia un gen por otro de manera aleatoria, similar a como ocurre en los seres vivos.

5 Propuesta

Ahora que se han establecido los fundamentos teóricos de los algoritmos genéticos, es importante definir los métodos específicos que implementados en este proyecto.

5.1 Inicialización de la población

Con la representación de los individuos previamente definida, la población inicial se genera creando claves con letras aleatoriamente seleccionadas del abecedario estándar (excluyendo caracteres especiales y espacios). La longitud de estas claves se establece en un rango entre 2 y un tercio de la longitud del texto plano a cifrar, una constante predefinida. Esta combinación de clave y texto plano se utiliza para cifrar el mensaje, dando como resultado la creación de un individuo, es decir, una instancia de una clase.

5.2 Selección de población

A pesar de probar distintos métodos de selección durante distintas pruebas en el código fuente del proyecto, parece que la función adecuada para este mismo es la *selección por torneos*, dónde, se eligen subgrupos de individuos de la población, y los miembros de cada subgrupo compiten entre ellos. Sólo se elige a un individuo de cada subgrupo para la reproducción.

5.3 Función de aptitud

La dos funciones propuestas en el apartado 3.2 son probadas en el *mismo* código, con un par de variantes:

- La utilización del método de Kasiski implica una maximización de la función de aptitud, por lo que, en el método de selección por torneos, se eligen a los individuos con *mayor* fitness.
- Por otro lado, con el uso del conteo máximo de incidencias, se busca encontrar a los individuos con el *menor* fitness.

5.4 Reproducción de los individuos

Debido a la diversidad de longitudes de los individuos, parece coherente elegir una función de cruce en la que no se suponga una longitud igual entre los padres. Por lo tanto, el cruce de un punto parece adecuado. En este método, se establece un punto de intercambio en un lugar aleatorio del genoma de los dos individuos, y uno de los individuos contribuye todo su código anterior a ese punto, mientras que el otro individuo contribuye todo su código a partir de ese punto para producir una descendencia. Así, el punto de cruce está dado por la longitud mínima de los padres, para asegurarse de que el tamaño de los hijos no exceda el tamaño de alguno de sus padres, y se sigan creando individuos dentro del rango establecido inicialmente.

5.5 Mutación de los individuos

Para mutar a los individuos, se utilizará una función llamada *mutación por intercambio*. Como su nombre lo indica, en esta función se intercambian dos letras en posiciones aleatoriamente elegidas de un solo individuo.

6 Resultados y discusión

Para evaluar el desempeño del proyecto con distintas funciones de aptitud, se procesó un texto suficientemente extenso pero manejable: las primeras 3 cuartillas de la novela "Moby Dick" de Herman Melville (1851), que, tras eliminar signos de puntuación y espacios, resultó en 4691 caracteres.

Es necesario dibujar una línea entre los resultados obtenidos durante las pruebas con la función de aptitud del método Kasiski, y la función del conteo de incidencias máxima.

6.1 Método Kasiski

Durante las pruebas con la función de aptitud basada en el método Kasiski, los resultados no cumplieron con las expectativas. Como se explicó anteriormente, el método de Kasiski tiende a fallar en ciertas condiciones específicas, como la necesidad de un texto cifrado de longitud significativa en relación con la longitud de la clave, y la presencia de patrones distintivos para realizar el criptoanálisis.

Se programó que el método de Kasiski devolviera un valor de fitness de 0 para los individuos de los que no se pudo aproximar una longitud de clave. Sin embargo, independientemente de diversos factores como la longitud de la población, la longitud de los individuos y el texto cifrado, este enfoque determinó que entre el 60 y 80 por ciento de la población era imposible de aproximar en términos de longitud de clave de cifrado.

A pesar de que inicialmente podría parecer ventajoso que la mayoría de los textos cifrados resultaran imposibles de romper con el método de Kasiski, esto generó dificultades en el algoritmo genético. La incapacidad para distinguir entre los individuos con fitness 0 llevó a una convergencia prematura y a una selección repetitiva de valores que el método de Kasiski no pudo aproximar.

6.2 Conteo máximo de incidencias

Por otro lado, los resultados obtenidos durante las pruebas del método de incidencias máximas fueron mucho más alentadores. Al tratarse de una función de minimización, se esperaba que los individuos con menor fitness fueran mejores que los demás.

Como se puede observar en la gráfica de la aptitud promedio de los individuos, el fitness promedio disminuyó con cada generación, lo que indica un

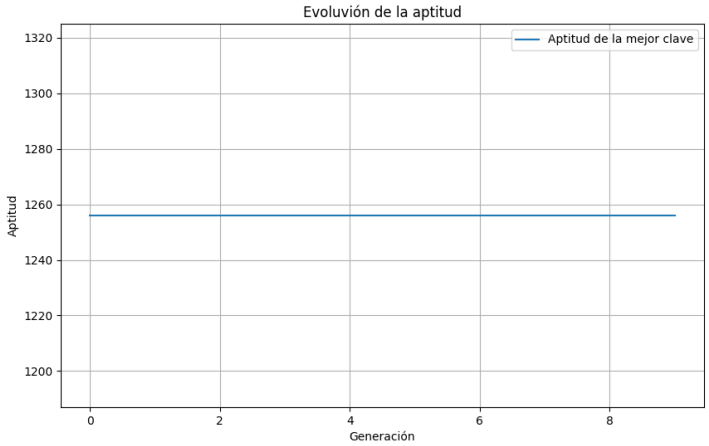

Figure 3: Ejecución del algoritmo utilizando el método Kasiski

progreso significativo hacia el objetivo inicial.

Estas pruebas indican que esta función es más adecuada para modelar el problema planteado desde el principio, y ofrece varias ventajas sobre el método de Kasiski:

- Este método, no necesita de una longitud de texto cifrado grande, en realidad, funciona bien con textos cortos, con menos de 200 caractéres.
- Las longitudes de las claves variables no representan un problema tan grande como para Kasiski, para este método resultan cobrar menos importancia.
- No falla, por lo que no es necesario cubrir los casos en los que lo hace, y se debe dar un peso mayor a estas soluciones.

Es importante destacar que el método de Kasiski presenta una complejidad temporal mucho menor que el método de conteo máximo de incidencias. A pesar de esto, el método de conteo máximo de incidencias sigue siendo preferible debido a su efectividad y eficiencia en la optimización de claves.

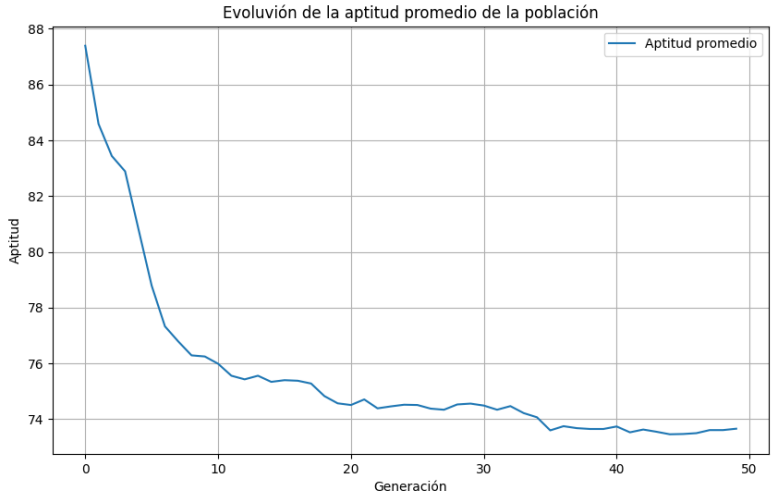

Figure 4: Aptitud promedio de los individuos con las generacións utilizando el método de incidencias máximas

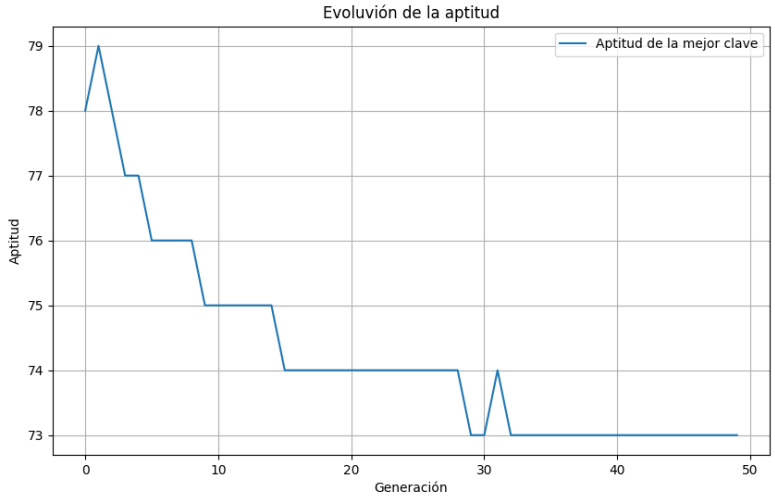

Figure 5: Aptitud de los mejores individuos a lo largo de las generaciones usando el método de incidencias máximas

7 Conclusiones

Durante el desarrollo de este proyecto, se exploraron y evaluaron diferentes enfoques para resolver el problema de optimización de la clave de encriptación en el cifrado de Vigenère. A través de la implementación de algoritmos genéticos y la utilización de funciones de aptitud específicas, se logró un mayor entendimiento de las complejidades involucradas en el criptoanálisis y la optimización de claves.

Uno de los hallazgos más significativos fue la **comparación** entre la función de aptitud basada en el método de Kasiski y la función de conteo de incidencias máximas. Si bien el método de Kasiski es ampliamente utilizado en criptoanálisis, sus limitaciones en este contexto específico fueron evidentes. La incapacidad para aproximar la longitud de la clave de manera efectiva y las dificultades para gestionar casos de fallo demostraron ser desafíos significativos en la implementación del algoritmo genético.

Por otro lado, la función de conteo de incidencias máximas mostró ser más efectiva y eficiente en la optimización de claves. Su capacidad para mejorar la aptitud de los individuos de manera consistente a lo largo de las generaciones sugiere que este enfoque es más adecuado para modelar el problema planteado. Además, su simplicidad y robustez lo convierten en una herramienta valiosa para resolver problemas de criptoanálisis mediante algoritmos genéticos.

En última instancia, este proyecto demostró que, la utilización de funciones de aptitud específicas puede tener un impacto significativo en el rendimiento y la eficacia de los algoritmos genéticos. Resulta casi increíble el saber que una función general resultó modelar el problema de una manera más adecuada que los métodos tradicionales. Esto subraya la importancia de la adaptabilidad y la exploración en el campo de la optimización y el criptoanálisis.

References

[1] David Arboledas Brihuega. *Criptografía sin secretos con Python*. RA-MA Editorial, Paracuellos de Jarama, Madrid, 2017.

[2] Fernando Sancho Caparrini. Algoritmos genéticos, 2024. Profesor Contratado Doctor, CCIA, Universidad de Sevilla.

[3] Isabel María Plaza del Pino and et al. La evaluación formativa y compartida como aspecto fundamental en la formación del profesorado novel universitario: experiencia en las facultades de ciencias y farmacia de la universidad de granada. *Psychology, Society & Education*, 4(1):59–72, 2012.

[4] Shivanshu. Vigenère cipher - the complete guide with examples, 2024. 26899 Views.