Integrantes del equipo:

Alejandro Chávez Mier 173199

Héctor Alejandro Vela Toriz 149427

Manuel García Garduño 162136

Karla Susana Alva Ortiz 174144

Código implementado en Python.

El objetivo es crear una clase, dentro de la cual se tengan los métodos de BFGS, algoritmo de Newton y Búsqueda Lineal de Newton con modificación a la Hessiana, además de las funciones necesarias dentro de estos algoritmos, tales como el gradiente y la Hessiana. De este modo, dada cualquier función, y un punto inicial, podemos encontrar el óptimo de la función con el método que se desee. Para la base de datos de crímenes elegimos el delito de secuestro que cuenta con 22 registros.

Descripción general:

1.- Clase Optimizador permite optimizar una función a partir de los métodos BFGS, algoritmo de Newton o Búsqueda Lineal de Newton con modificación a la Hessiana. Para estos métodos se creó una función que calcula el gradiente, la Hessiana y el tamaño del paso.

2.- Se definió la función de Rosenbrock, con la cual se probaron los algoritmos de la Clase Optimizador. Se conoce que su óptimo se encuentra en (a, a^2).

3.- Función de costo, se realiza la lectura de los datos de los crímenes, y calcula el costo de la posición de las cámaras.

4.- Se seleccionaron los crímenes de secuestro, y se buscó con los diferentes métodos de Clase Optimizador la mejor posición para 5 cámaras alrededor de CDMX.

Algunos detalles:

1.- Para el método BFGS fue necesario introducir una tolerancia para el producto punto yk\*sk, ya que antes de se cumpliera con la restricción de que la norma del gradiente fuera menor que épsilon, yk\*sk se hacía cero, lo que indefinía el valor de ro. En este caso, cuando se llegaba a la tolerancia de yk\*sk, en automático se termina el método.

2.- Después de la lectura de los datos de los crímenes, estos se transformaron de un vector a una matriz, en donde la primera columna corresponde a la longitud, y la segunda a la latitud.

3.- El vector inicial de la posición de las cámaras se creó aleatoriamente, donde los valores de las latitudes oscilan entre los 19 y 20 grados, y los valores de las longitudes entre los -99 y -100 grados. De esta manera, la posición inicial de las cámaras se encuentra dentro de la CDMX. La función de **costo** recibe como parámetro a un vector que indica las coordenadas de cada una de las cámaras. La primera mitad del vector son las latitudes y la segunda mitad son las longitudes. Por ejemplo, si el vector de input en la función **costo** tiene 100 entradas, entonces la entrada 0 y la entrada 50 corresponden a la latitud y longitud de a primera cámara.