Practica No. 2

Métodos numéricos de optimización multidimensional no restringida Método de Búsqueda Aleatorio

Nombre(s):

César Eduardo Elías Del Hoyo

Jose Luis Sandoval Perez

*Objetivo:*

Con la realización de esta práctica se pretende: que el alumno implemente un programa que utilizando el método de Newton resuelva un problema de optimización multidimensional no restringida.

*Fundamento Teórico:*

dimensión.

multivariables) involucran la resolución de un problema de optimización matemática en una

Muchos métodos de optimización de problemas con restricciones (univariables y

**Optimización**

intuitivas.

puede influir o sesgar la estrategia de optimización, lo que permite descubrir soluciones no

Esto puede ser beneficioso para problemas en los que hay mucha experiencia en el dominio que

Es eficaz porque no asume nada sobre la estructura de la función objetivo.

La búsqueda aleatoria implica generar y evaluar entradas aleatorias a la función objetivo.

**Método de Búsqueda Aleatoria**

La búsqueda aleatoria también puede ser la mejor estrategia para problemas muy complejos

con áreas ruidosas o no uniformes (discontinuas) del espacio de búsqueda que pueden generar algoritmos que dependen de gradientes confiables.

Se puede generar una muestra aleatoria de un dominio usando un generador de números pseudoaleatorios.

Cada variable requiere un límite o rango bien definido y se puede muestrear un valor aleatorio uniforme del rango y luego evaluarla.

Generar muestras aleatorias es sencillo desde el punto de vista computacional y no ocupa mucha memoria, por lo tanto, puede ser eficiente generar una muestra grande de entradas y luego evaluarlas. Cada muestra es independiente, por lo que las muestras se pueden evaluar en paralelo si es necesario para acelerar el proceso.

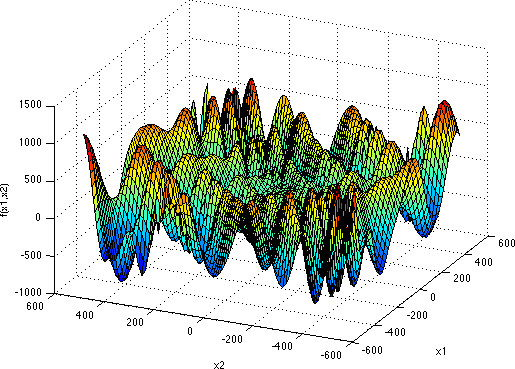
*Forma de trabajo:*

Colaborativa en equipos de 3 personas

*Material:*

1. Computadora
2. Compilador C/C++

*Procedimiento:*



1. Minimizar la siguiente función utilizando el Método de Búsqueda Aleatoria

**EGGHOLDER FUNCTION**

1. Como puede apreciarse, la función Eggholder es una función difícil de optimizar debido a la

gran cantidad de mínimos locales.

1. Dominio de entrada: La función generalmente se evalúa sobre el cuadrado xi ∈ [-512, 512], para todo i = 1, 2.
2. Determinar la condición de paro de la búsqueda

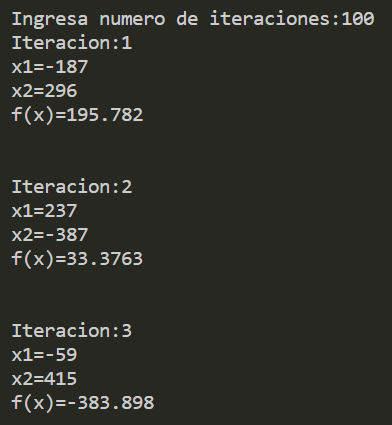
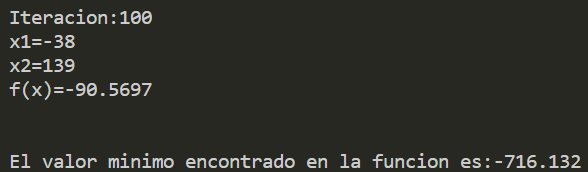
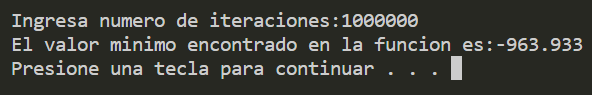
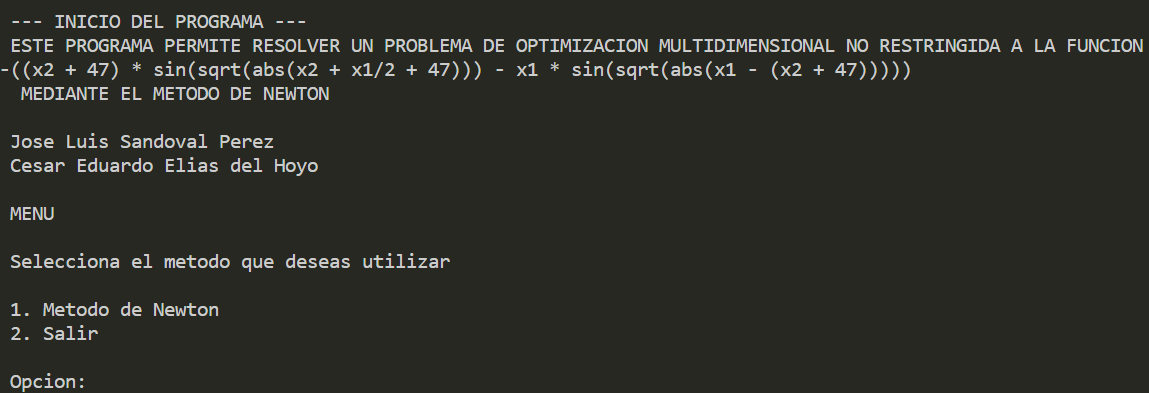
Para la creación del programa deberán realizarse los siguientes pasos:

1. En las primeras líneas elaborar comentarios con la siguiente información:
   1. Nombre de la institución
   2. Nombre del centro al que pertenece la carrera
   3. Nombre del departamento al que pertenece la carrera
   4. Nombre de la materia

*Resultados:*

1. Nombre(s) de quien(es) realiza(n) la práctica
2. Nombre del profesor
3. Una descripción breve de lo que realiza el programa
4. Incluir las librerías necesarias.
5. Iniciar la generación de aleatorios dentro del rango.
6. Evaluar para cada aleatorio y determinar el valor mínimo de la función original.
7. Se debe desplegar en la pantalla los valores de no\_iteración, xi, f(x), en una tabla que muestre los valores de estas variables durante cada iteración.
8. Desplegar el valor mínimo encontrado de la función.
9. Una vez realizada cualquier operación se debe regresar al menú principal.
10. Al salir se debe detener el programa y luego regresar el control al sistema inicial.

En los siguientes cuadros copiar las pantallas de 2 corridas con valores de inicio diferentes del programa.

**

*Conclusiones:*

*El metodo de newton y de búsqueda aleatoria es muy útil para la resolución de este tipo de problemas, la funcion EggHolder a pesar de ser muy complicada de optimizar con este metodo pudimos aproximar un buen mínimo aproximado. Sin duda el criterio de paro limitando las iteraciones fue de gran ayuda.*