



Instituto Politecnico Nacional

Escuela Superior de Computo

Ingeniería en Sistemas Computacionales

Sexto semestre

Grupo: 6CV3

## **Inteligencia Artificial**

### **Practica 5: Función Himmelblau**

Presentado por:

Flores Vicencio Marco Isaí

Pérez Gonzalez Jesús

Reyes Núñez Sebastián

Fecha: 08/10/2024



## Indice

<i>Introducción</i> .....	3
<i>Desarrollo</i> .....	4
<i>Conclusiones</i> .....	6
<i>Referencias</i> .....	6



### ***Introducción***

La función de Himmelblau es una función multimodal que se usa en la optimización matemática para comprobar el rendimiento de los algoritmos de optimización.

Fue propuesta por el estadounidense David Mautner Himmelblau. Es una función definida en un espacio  $\mathbb{R}^2$  que permite obtener parámetros de algoritmos de optimización como ratio de convergencia, precisión y robustez.

Su representación matemática es mostrada como una función de dos variables tal que:

$$f(x, y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2$$

Manteniendo su dominio en todos los reales dentro del intervalo  $[-6, 6]$

La función Himmelblau es conocida por tener valores de  $x, y$  para hallar cuatro mínimos locales y un máximo local, representados de la siguiente forma:

$$f(x, y) = 0 \text{ con } (x, y) = 3, 2$$

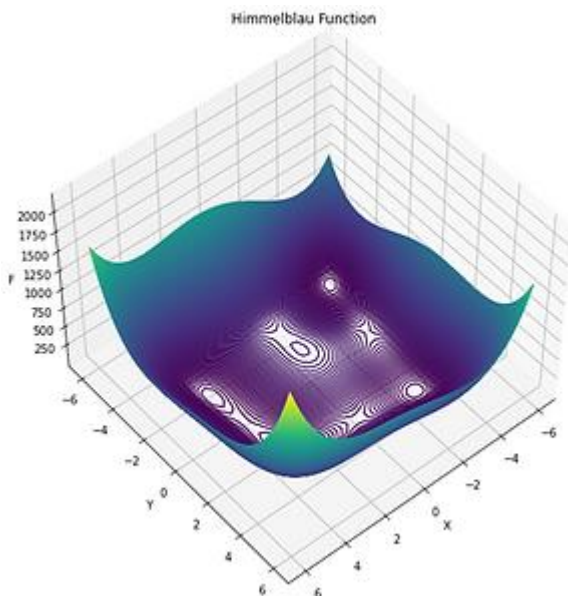
$$f(x, y) = 0 \text{ con } (x, y) = -2.805118, 3.283186$$

$$f(x, y) = 0 \text{ con } (x, y) = -3.779310, -3.283186$$

$$f(x, y) = 0 \text{ con } (x, y) = 3.584458, -1.848126$$

$$\text{Máx } f(x, y) = 181.617 \text{ con } (x, y) = -0.270845, -0.923039$$

Finalmente, la función Himmelblau puede ser graficada de la siguiente forma:





## Desarrollo

Para el desarrollo de esta práctica, se utilizó un algoritmo para encontrar los valores donde la función de Himmelblau es mínima. Para hallar estos valores, se toma un dominio de  $[-5, 5]$  para los valores de  $x, y$ . Se utiliza un algoritmo de búsqueda aleatoria en diez mil iteraciones para encontrar los cuatro mejores valores.

```
import numpy as np #Importar biblioteca para manejar expresiones matematicas complejas
```

Importamos la biblioteca *numpy*, ya que se encarga de manejar expresiones matemáticas complejas y realizar operaciones sobre esta con mayor facilidad.

```
def himmelblau(x, y): #Definir metodo para la función Himmelblau
    return (x**2 + y - 11)**2 + (x + y**2 - 7)**2
```

Se define un método que almacena la función de Himmelblau, misma que será evaluada con los valores de  $x, y$  encontrados en las iteraciones aleatorias.

```
def random_search(iterations=10000): #Método para búsqueda aleatoria con 10,000 iteraciones
    best_values = [] #Lista para almacenar los mejores valores encontrados en las 10,000 iteraciones
    for _ in range(iterations):
        x = np.random.uniform(-5, 5) #Generación de números aleatorios para variable en un intervalo de -5 a 5
        y = np.random.uniform(-5, 5)

        f_val = himmelblau(x, y) #Evalua función

        best_values.append((x, y, f_val)) #Almacenar los mejores valores

    best_values.sort(key=lambda point: point[2]) #Ordenamiento para los valores

    best_4 = best_values[:4] #Toma los mejores 4 de la lista

    return best_4
```

Se define un método *random\_search* que, como su nombre indica, se usará para utilizar un algoritmo de búsqueda aleatoria en diez mil iteraciones.

Se crea una lista donde se almacenarán los valores encontrados en las iteraciones. Mediante un ciclo *for* se crean valores para  $x, y$  en intervalos de  $-5$  a  $5$ . Para cada iteración se toman los valores y se evalúan en la función definida en el método *himmelblau*. Finalmente, el valor se almacena en la lista.

Al término de las iteraciones, se realiza un ordenamiento para tomar los cuatro mejores valores de la lista, mismos que serán comparados con los valores obtenidos en la investigación previa.



```
def main():
    best_4_minima = random_search()

    print("Los mejores 4 valores donde la función es mínima:")
    for i, (x, y, f_val) in enumerate(best_4_minima, 1):
        print(f"{i}: x = {x:.6f}, y = {y:.6f}, f(x,y) = {f_val:.6f}")

if __name__ == "__main__":
    main()
```

Finalmente, se llama al método *random\_search* y se imprimen los valores obtenidos. Estos son mostrados como valores de punto flotante con seis decimales, además de mostrar el valor de la función que se asemeja al mínimo.

### Resultados

```
P5 C:\Users\marco\Desktop> & C:/Users/marco/AppData/Local/Programs/Python/Python311/python.exe c:/Users/marco/Desktop/Himmelblau.py
Los mejores 4 valores donde la función es mínima:
1: x = 2.977829, y = 2.011271, f(x,y) = 0.016378
2: x = -2.829412, y = 3.147574, f(x,y) = 0.029588
3: x = 3.594267, y = -1.792976, f(x,y) = 0.052289
4: x = -3.782810, y = -3.247159, f(x,y) = 0.059816
```

Como se puede observar, se muestran los cuatro mejores resultados al ejecutar el código. Estos valores nos sirven para conocer el valor mínimo de la función.



### ***Conclusión***

El desarrollo del programa para la función de Himmelblau a través de un algoritmo de búsqueda aleatoria representa un enfoque práctico para aplicar los conceptos teóricos de optimización en un entorno computacional. Esto nos permitió comprender mejor cómo funcionan las funciones multimodales y cómo se pueden aplicar técnicas de búsqueda para encontrar soluciones óptimas, incluso en problemas no triviales.

La función de Himmelblau, al ser multimodal, es ideal para estudiar los desafíos de la optimización, como la convergencia hacia mínimos locales y la eficiencia de los algoritmos en términos de precisión y robustez. El uso de la búsqueda aleatoria es un enfoque simple, pero efectivo, que resalta la importancia de técnicas más avanzadas como los algoritmos genéticos o los gradientes, los cuales podrían ser explorados en estudios más avanzados.

Finalmente, este tipo de prácticas integran tanto teoría matemática como habilidades de programación, lo que resulta crucial en nuestra formación, y que nos servirá para seguir enfrentando este tipo de problemas, ya sea en el ámbito escolar o profesional.

### ***Referencias***

- Limited, I. P. (2021, July 4). HIMMELBLAU FUNCTION. *Indusmic*.  
<https://www.indusmic.com/post/himmelblau-function>
- *Minimization of the Himmelblau Function — algopy documentation*. (n.d.).  
[https://pythonhosted.org/algopy/examples/minimization/himmelblau\\_minimization.html](https://pythonhosted.org/algopy/examples/minimization/himmelblau_minimization.html)
- Flores, M. Á. (2021, December 10). Graficando la función de Himmelblau - Miguel Ángel Flores - Medium. *Medium*.  
<https://medium.com/@hdezfloresmiguelangel/graficando-la-funci%C3%B3n-de-himmelblau-6045c4b865f>